

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**Desenvolvimento de um Gerador de “Scheduling” para
uma Indústria de Produção sob Encomenda: Uma
Abordagem Baseada no Uso de Controladores Difusos e
Algoritmos Genéticos**

Fernando Martineli Loureiro

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Produção
como requisito à obtenção do grau de Mestre em Engenharia



0.236.149-1

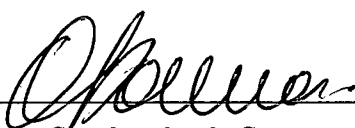
UFSC-BU

Florianópolis, junho de 1995

Desenvolvimento de um Gerador de “Scheduling” para uma Indústria de Produção sob Encomenda: Uma Abordagem Baseada no Uso de Controladores Difusos e Algoritmos Genéticos

Fernando Martineli Loureiro

Esta dissertação foi julgada adequada para obtenção do título de mestre em Engenharia de Produção e aprovada em sua forma final pelo Programa de Pós-graduação



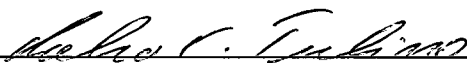
Coordenador do Curso
Osmar Possamai, Dr.

Banca Examinadora:



Orientador

Prof. Paulo José de Freitas Filha, Dr.



Prof. Dalvio Ferrari Tubino, Dr.



Prof. Cristiano J. C. de A. Cunha, Dr. rer. pol.

SUMÁRIO

Lista de Figuras	vi
Lista de Tabelas	viii
Resumo	ix
Abstract	x

CAPÍTULO I INTRODUÇÃO

1.1. Origem do Trabalho	1
1.2. Considerações Práticas no Projeto do Sistema	2
1.3. Definição do Problema	3
1.4. Objetivos do Trabalho	4
1.5. Importância do Trabalho	4
1.6. Suposições, Alcance e Limitações do Trabalho	5
1.7. Organização do Trabalho	6

CAPÍTULO II REVISÃO BIBLIOGRÁFICA

2.1. Planejamento da Operação	7
2.1.1. Controle da Operação	8
2.1.1.1. Despacho de Peças	9
2.1.1.2. Despacho de Veículos	12
2.1.2. Impacto dos Parâmetros do Sistema	14
2.2. Aplicações de Inteligência Artificial para Otimização de <i>Scheduling</i>	15
2.3. Sumário	18

CAPÍTULO III PRINCIPAIS FERRAMENTAS UTILIZADAS

3.1. Simulação	19
3.1.1. Vantagens e Desvantagens do Uso da Simulação	21
3.1.2. O Processo de Simulação	23
3.1.3. Aplicação de Simulação em Processos de <i>Scheduling</i>	24

SUMÁRIO

3.2. Sistemas Difusos	25
3.2.1. Regras Difusas	26
3.2.2. Tomada de Decisão Multi-Atributo Difusa	28
3.3. Algoritmos Genéticos	30
3.3.1. Algumas aplicações dos GAs	33
3.3.2. Comparação da terminologia da Genética e dos GAs	35
3.4. Sumário	36

CAPÍTULO IV	<i>O MODELO PROPOSTO PARA A GERAÇÃO DE SCHEDULING</i>
--------------------	--

4.1. Razões para desenvolvimento de um gerador de <i>scheduling</i>	37
4.2. Elementos do Planejamento da Produção	39
4.3. Elementos do Controle Operacional	40
4.3.1. Regras para a Liberação de Novas Peças ao Sistema	41
4.4. O <i>Scheduling</i>	43
4.5. O Gerador de <i>Scheduling</i> Proposto	44
4.6. A Simulação	47
4.6.1. Projeto do Controlador	47
4.6.1.1. Controlador de Fila	48
4.6.1.1.1. Regras para a Priorização de Peças no <i>Buffer</i> das Máquinas	50
4.6.1.2. Processadores de Fila	51
4.6.1.3. Controladores e Processadores de Transporte	52
4.6.1.3.1. Políticas para o Despacho de Veículos Transportadores	54
4.7. Parâmetros Utilizados no Modelo	54
4.8. Modelo de Simulação Proposto	56
4.8.1. Hipóteses Associadas ao Modelo	57
4.8.2. Controle Lógico das Máquinas	57
4.8.3. Medidas de Desempenho	58
4.9. Sumário	63

CAPÍTULO V

APLICAÇÃO DO MODELO PROPOSTO

5.1. Empresa Escolhida	64
5.2. Modelagem do Sistema Produtivo	66
5.2.1. Coleta e Análise de Dados	66
5.2.2. Análise de Fluxos	66
5.3. Modelagem do Sistema	67
5.3.1. Características do Modelo	67
5.4. Desenvolvimento do Modelo de Simulação no ARENA	69
5.5. Experimento	72
5.6. Resultado da Simulação	75
5.7. Comparação entre Modelo Proposto e Aplicação Pura das Regras	84
5.8. Sumário	87

CAPÍTULO VI

CONCLUSÕES E RECOMENDAÇÕES

6.1. Resultados e Contribuições da Pesquisa	88
6.2. Limitações do Trabalho	90
6.3. Recomendações para Futuras Pesquisas	91
6.3.1. Desenvolvimento de novos controladores	91
6.3.2. Aplicação em Ambientes Tipo <i>Flow-Shop</i>	92
6.3.3. Aplicação de Mais Regras nos Controladores	92
6.3.4. Aperfeiçoamento do Controlador Difuso	92

ANEXOS

ANEXO I	Listagem do Modelo de Simulação	93
ANEXO II	Listagem dos GAs	100
ANEXO III	Glossário dos Principais Conceitos de Genética Aplicados aos GAs	107
ANEXO IV	Roteiro e Tempo de Fabricação	110
ANEXO V	Dependências	116

REFERÊNCIAS BIBLIOGRÁFICAS

120

LISTA DE FIGURAS

Figura 2.1.	Fluxo de informações do modelo proposto por Cho e Wysk	25
Figura 3.1.	Três subconjuntos difusos do espaço X	32
Figura 4.1.	Elementos dos Modelos de Planejamento	48
Figura 4.2.	Elementos do Controle Operacional	50
Figura 4.3.	Esquema de funcionamento do gerador de <i>scheduling</i>	54
Figura 4.4.	Esquema de ligação dos sistemas que compõe o modelo proposto	55
Figura 4.5.	Representação esquemática do controlador de fila	58
Figura 4.6.	Lógica de funcionamento do controlador de fila e do processador central associado	59
Figura 4.7.	Cálculo das pertinências	60
Figura 4.8.	Representação esquemática do controlador de transporte	60
Figura 4.9.	Lógica de funcionamento do processador de transporte	61
Figura 4.10.	Controle Lógico das Máquinas	65
Figura 4.11.	Esquema de funcionamento dos GAs	77
Figura 4.12.	Representação esquemática dos operadores genéticos usados	78
Figura 5.1.	Layout da fábrica	69
Figura 5.2.	Evolução da população em termos do <i>fitness</i> , avaliados por valores médios e máximos durante a calibração do controlador de fila para condição de minimização do tempo médio em atraso.	81
Figura 5.3.	Evolução da população em termos do <i>fitness</i> , avaliados por valores médios e máximos durante a calibração do controlador de fila para condição de minimização do tempo máximo em atraso.	82
Figura 5.4.	Evolução da população em termos do <i>fitness</i> , avaliados por valores médios e máximos durante a calibração do controlador de transporte para condição de minimização do tempo médio em atraso.	82

LISTA DE FIGURAS

- Figura 5.5. Evolução da população em termos do *fitness*, 83
avaliados por valores médios e máximos durante a
calibração do controlador de transporte para
condição de minimização do tempo máximo em
atraso.
- Figura 5.6. Curva de resposta para o controlador de fila 88
calibrado para minimização do tempo médio em
atraso.
- Figura 5.7. Curva de resposta para o controlador de fila 89
calibrado para minimização do tempo máximo em
atraso.
- Figura 5.8. Curva de resposta para o controlador de transporte 89
calibrado para minimização do tempo médio em
atraso.
- Figura 5.9. Curva de resposta para o controlador de transporte 90
calibrado para minimização do tempo máximo em
atraso
- Figura 5.10 Gráfico de desempenho em termos de atraso
médio para cada regra utilizada pelo controlador
de fila e pela aplicação dinâmica das mesmas
através do modelo proposto
- Figura 5.11 Gráfico de desempenho em termos de atraso
máximo para cada regra utilizada pelo controlador
de fila e pela aplicação dinâmica das mesmas
através do modelo proposto
- Figura 5.12. Gráfico de desempenho em termos de atraso
médio para cada regra utilizada pelo controlador
de transporte e pela aplicação dinâmica das
mesmas através do modelo proposto
- Figura 5.13. Gráfico de desempenho em termos de atraso
máximo para cada regra utilizada pelo controlador
de transporte e pela aplicação dinâmica das
mesmas através do modelo proposto

LISTA DE TABELAS

Tabela 2.1	Termos alterados pela teoria dos GAs em relação à genética	35
Tabela 5.1	Parâmetros iniciais utilizados	72
Tabela 5.2	<i>Strings</i> possíveis, seus respectivos códigos decodificadores, o valor associado aos parâmetros difusos dos controladores	74
Tabela 5.3.	Resultados da simulação	79
Tabela 5.4.	Resultados da simulação	80
Tabela 5.5.	Valores obtidos para os parâmetros difusos após a calibração do controlador de fila	81
Tabela 5.6.	Valores obtidos para os parâmetros difusos após a calibração do controladpr de transporte	81
Tabela 5.7	Atraso médio e máximo atraso para cada regra utilizada pelo controlador de fila	84
Tabela 5.8.	Atraso médio e máximo atraso para cada regra utilizada pelo controlador de transporte	85

Geralmente, os administradores tendem a se concentrar apenas no projeto relacionados ao chão de fábrica para melhorar o desempenho de suas indústrias. Contudo, estes têm descoberto que apenas com o planejamento visando a integração dos recursos disponíveis é que isto será possível. Este trabalho pretende, portanto, desenvolver uma nova abordagem para responder a este desafio.

Um sistema de controle do chão de fábrica, consistindo de dois níveis de controle hierárquicos (controladores de chão de fábrica e máquina) é proposto. Procurou-se, através deste, gerar *scheduling* em tempo real, a fim de executar as atividades necessárias ao processo de uma ordem de fabricação. O sistema proposto para alcançar a melhor programação é baseado na aplicação de sistemas difusos, algoritmos genéticos e no uso da simulação de processos.

A necessidade de utilização de uma abordagem que leve em consideração os aspectos dinâmicos do sistema é resultado do fato de que a realidade não pode ser completamente planejadas, i.e, eventos não podem ser tratados a priori.

O método de otimização através de algoritmos genéticos (GAs) são baseados nos mecanismos de seleção natural e na genética. Os GAs exploram informações históricas para especular sobre novos pontos, onde sejam esperados aumento de desempenho do sistema. Aqui são utilizados para ajustar parâmetros difusos dos controladores utilizados para despacho de peças.

O sistema foi testado em uma indústria que produz vários tipos de produtos em pequenos volumes. Esta planta é capaz de produzir por volta de 1000 peças componentes de 100 produtos disponíveis. É importante salientar que neste tipo de sistema de produção, onde cada tarefa possui um roteiro diferente, um eficiente sistema gerencial é extremamente requerido. Neste caso, regras simples foram comparadas com a seleção dinâmica de regras. Verificou-se pela simulação do *scheduling*, que este sistema apresentou-se efetivo no decréscimo do tempo médio e máximo em atraso relativo à data de entrega.

Traditionally managers concerning on design individual production shops to improve the process. But, manufactures are becoming aware that the major gains in competitiveness made necessary by the growing challenge of global markets, are available only from integrated production planning. This work aims to answer this question by developing a new model to solve this new challenge.

A shop floor control system consisting of two hierarchical control levels (shop and machine control) is proposed. It aims to produce on-line scheduling to execute the activities necessary to process an order. A system based on Simulation, Genetic Algorithms and Fuzzy Controllers to find the best schedule order is proposed.

The dynamical approach is necessary because usually the real situation cannot be planned, i.e., some events cannot be treated before they arise. These events have notable influence on the systems performance, and they cannot be disregarded.

The Genetic Algorithms, a powerful optimization algorithm, are search algorithms based on the mechanics of natural selection and natural genetics. They exploit historical information to speculate on new search points with expected improved performance. It is applied in this system to tune the fuzzy parameters of fuzzy controllers, that are used to control the despatch of parts.

The system was tested in a real industry that produces various kind of product in small volumes. This plant can produce around 1000 parts of 100 products. The throughput is around 200 products a year. It is important to point out that in this kind of job shop type production, in which each job has a different work-flow, efficient management is extremely required. The single rules are compared with this dynamical choosing process. The proposed system is verified by scheduling simulation to be effective in decreasing maximum lateness to due-date as much as average lateness.

1.1. Origem do Trabalho

Observa-se com frequência que a tentativa de empregar regras de *scheduling* utilizadas em ambientes diferentes gera, em muitos casos, conflitos que são de difícil solução. A razão para isto é que cada empresa possui uma única maneira de integrar recursos, transportes, sistemas de movimentação de materiais, lógicas de controle, políticas operacionais, além da determinação de mix de produção [PRISTKER, 1992]. Não é difícil, apesar disto, verificar que a cada dia na literatura diferentes heurísticas têm sido desenvolvidas, mas estas continuam a voltar-se para problemas específicos das plantas analisadas. A experiência prática sugere, portanto, que estas regras não são consistentes na aplicação geral.

Um sistema de *scheduling* da produção baseado na utilização de simulação possui vantagens evidentes, uma vez que permite representar com facilidade, ao nível de detalhe pretendido, todas estas restrições inerentes aos diferentes sistemas, ao mesmo tempo em que concentra no modelo de simulação as características próprias de cada planta, isto é, permite que as alterações sejam feitas apenas no modelo de simulação. Por esta razão, a aplicação conjunta da simulação com sistemas do tipo MRP e MRPII, controladores de chão-de-fábrica e sistemas de *scheduling* tem se tornado cada vez mais freqüente, como pode ser visto nos trabalhos de Bilberg e Alting [1992] e no de Cho e Wisk [1993]. Neste tipo de aplicação, a simulação exerce a função central do sistema, interconectando todas as ferramentas envolvidas, isto é, o simulador recebe entradas (alternativas) e fornece saídas

Introdução

(resposta do sistema para aquelas alternativas), proporcionando assim dados para a tomada de decisão [NORMAN, 1992].

A necessidade de desenvolvimento de um sistema para programação da produção de um sistema produtivo trabalhando sob encomenda, onde alterações de *layout* e de produtos estavam em andamento, levou ao desenvolvimento de um sistema flexível o bastante para absorver tais modificações.

Desenvolveu-se, aqui, um sistema de auxílio, que possibilita a inclusão do maior número de características relevantes aos sistemas, visando a determinação do seqüenciamento das peças nas filas, isto é, designando para cada peça a prioridade de entrada para processo e para transporte.

1.2. Considerações Práticas no Desenvolvimento do Projeto

O problema real no desenvolvimento de um sistema de *scheduling* não se refere a questões de estabelecer fundamentações teóricas corretas, mas na necessidade de prover uma solução aplicável e satisfatória. Isto significa que se deve levar em consideração tanto o planejamento de médio e longo prazo, quanto os requisitos necessários ao gerenciamento do dia-a-dia da empresa. Faz mister salientar, ainda, que qualquer tentativa de solução deva ser evolucionária, atendendo às modificações feitas na empresa, no mercado e nos fornecedores, sempre tendo em vista que o desenvolvimento de aplicações específicas muitas vezes tornam soluções obsoletas ainda antes de sua implantação efetiva.

O modelo, ora proposto, visa flexibilizar a estrutura do processo de tomada de decisão, através do uso de um modelo de simulação. Este modelo possibilita que as restrições impostas pelo ambiente, ou que modificações efetuadas

possam ser repassadas ao sistema, mais especificamente ao modelo de simulação, sem implicar necessariamente na modificação geral do sistema. Este sistema tende assim a ser robusto, evitando o risco de uma obsolescência precoce.

1.3. Definição do Problema

O nível das soluções colocadas a nível operacionais nas plantas industriais são em geral bastante pobres e/ou ineficientes. Isto se deve a dificuldade do decisor em lidar com uma vasta série de variáveis necessárias a tomada de decisão, uma vez que o excesso de alternativas e combinações destas inviabiliza a análise de todas as possibilidades disponíveis. Observa-se também que os pacotes computacionais disponíveis voltam-se sobretudo à fase de planejamento, deixando o controle operacional restrito às atividades de chão de fábrica, o que contribui para a desordem do sistema. Deve-se salientar ainda que alguns parâmetros relacionados à fase de operação do sistemas, tais como número de *pallets*, tamanho dos *buffers* e regras de despacho, são capazes de influenciar fortemente o desempenho do sistema, e até mesmo invalidar o planejamento feito *a priori*. Procura-se tratar, aqui, de forma mais cuidadosa a fase operacional, desenvolvendo um sistema computacional para a análise e tomada de decisão a este nível.

O caso aqui estudado se refere ao desenvolvimento de um sistema de controle da produção para uma indústria de produtos não seriados. Devido a característica de operação deste tipo de planta, onde há um número bastante elevado de produtos, cada qual com seu roteiro específico, seus tempos de máquina, seus componentes, mas com operação em lotes unitários, o controle operacional aumenta seu grau de importância, ao mesmo tempo que a maior complexidade do sistema aumenta as incertezas relacionadas ao nível operacional, diminuindo assim o número de decisões tomadas *a priori*, isto é, tomadas a nível de planejamento.

1.4. Objetivos do Trabalho

O objetivo principal deste trabalho é desenvolver e implementar um modelo que permita otimizar o desempenho de sistema de manufatura, considerando o inter-relacionamento de todas as características relevantes do sistema. É também objetivado a possibilidade de aplicação prática deste modelo nos mais variados tipos de indústria, o que implica sobretudo que este seja um modelo simples e flexível o bastante para permitir modificações.

Espera-se que com a retirada de algumas atribuições da fase de planejamento, e pela incorporação à fase operacional, uma melhora sensível no desempenho do sistema seja obtida. Acredita-se que através da monitoração do estado do sistema por meio de variáveis, as decisões tomadas sejam qualitativamente melhores.

1.5. Importância do Trabalho

Características próprias de cada sistema produtivo normalmente impossibilitam o desenvolvimento de pacotes computacionais de aplicação geral, auxiliares ao processo de elaboração do *scheduling*. Modificações nas plantas e, sobretudo, variações de demanda dificultam ainda mais este processo. Por isso, verificou-se a necessidade de desenvolver um método que tornasse possível a implantação de um gerador de *scheduling*, que fosse ao mesmo tempo de fácil modificação, a fim de se adequar rapidamente as mudanças, e potente o suficiente para gerar boas soluções para o chão de fábrica.

A importância deste trabalho baseou-se, sobretudo, no desenvolvimento de um gerador de *scheduling*, contribuindo para o uso da simulação como ferramenta de uso geral nas indústrias, particularmente na área de PCP (Planejamento e Controle da Produção).

1.6. Suposições, Alcance e Limitações do Trabalho

Algumas limitações e suposições iniciais são necessárias, a fim de tornar tratáveis os problemas envolvidos nos objetivos definidos neste trabalho, tanto do ponto de vista teórico como prático.

Todo sistema produtivo é único e depende de sua aplicação, de sua variedade de peças e produtos, da flexibilidade caracterizadas pelos seus equipamentos (*hardware* e *software*). Portanto torna-se impossível a criação de um modelo que permita a avaliação de qualquer sistema produtivo existente. Faz mister, então, definir primeiramente o ambiente a ser aplicado este modelo.

Escolheu-se, neste trabalho, o setor metal-mecânico, especificamente uma indústria produtora de peças por processos não contínuos do tipo *flow-shop*, operando sob encomenda. Neste ambiente é típico a presença de uma grande quantidade de produtos, bem como de uma demanda limitada. A complexidade das ordens de produção, a grande variedade de produtos, roteiro e tempos foram considerados como aparecem, sem que fosse feita nenhuma simplificação.

Assumiu-se, aqui, que problemas relativos a espaço de armazenagem, matéria-prima disponível e quebras de máquinas não teriam influência significativa sobre o processo. Alguns deste fatores foram parcialmente levados em consideração, através do acréscimo no tempo dos processos. Contudo sua implicação em relação ao fluxo e a lógica não puderam ser percebidos. Chama-se a atenção que o modelo desenvolvido permite que com a agregação de mais complexidade ao modelo de simulação proposto, estas simplificações possam ser substituídas por elementos mais detalhados. Neste trabalho procurou-se simplificar ao máximo

Introdução

a modelagem do sistema, pois o intuito do trabalho foi o de demonstrar a potencialidade das ferramentas e do método.

Assumiu-se também que o arranjo físico e as distâncias são conhecidas, sendo estes considerados fixos.

O alcance deste trabalho limita-se à determinação de um modelo de geração de *scheduling* aplicável em uma planta do tipo *job-shop* operando sob encomenda, considerando dinamicamente diversas estratégias de controle operacional.

1.7. Organização do Trabalho

Este trabalho se divide em seis capítulos. No primeiro comenta-se sobre os motivos que originaram a pesquisa, bem como define-se os problemas a serem tratados. Comenta-se, ainda, sua importância, limitações e alcance. No segundo capítulo, é feita uma revisão bibliográfica, através do levantamento do estado da arte dos diversos assuntos diretamente tratados e relacionados à pesquisa. No terceiro capítulo são descritas as principais ferramentas de AI utilizadas no gerador de scheduling proposto. No quarto capítulo é mostrado o modelo proposto, seguindo-se do capítulo que trata da aplicação prática do mesmo. No último capítulo são feitas considerações a respeito das conclusões obtidas, assim como são elaboradas recomendações sobre pesquisas futuras.

REVISÃO BIBLIOGRÁFICA

Este capítulo se dedica à descrição dos conceitos e teorias considerados fundamentais para o desenvolvimento do trabalho. Procurou-se através do levantamento do estado da arte da literatura existente conceituar os principais pontos, bem como descrever aplicações e pesquisas relacionadas a mesma área de aplicação.

2.1. Planejamento da Operação A operação de qualquer sistema produtivo requer uma cuidadosa programação das tarefas, envolvendo a disponibilidade de recursos, mão de obra e matéria-prima. Isto é feito visando obter uma utilização eficiente do sistema produtivo. No planejamento, uma das mais importantes funções é a da designação das máquinas responsáveis por determinadas operações. Para solucionar este tipo de problema, várias abordagens foram desenvolvidas, e as mais comuns são a programação linear inteira, a programação com múltiplos objetivos, a programação dinâmica e a teoria dos grafos [FREITAS, 1994].

Nos sistemas produtivos operando sob encomenda, o número de alternativas disponíveis para o tomador de decisão é consideravelmente menor, assim como é menor a incerteza associada ao processo. A diminuição da incerteza é diretamente relacionada ao problema de determinação da demanda, que neste caso é feita baseada nos pedidos feitos, ou seja o processo produtivo somente é desencadeado depois da confirmação da venda de um produto [LECOMTE e DEJAX, 1991]. Ainda, alterações no tamanho de lote ou no seqüenciamento da produção são relativamente restritos, isto é, não existem muitas possibilidades de modificação, uma vez que as peças são

Revisão Bibliográfica

fabricadas em lotes pequenos, muitas vezes até unitário. Contudo, ressalta-se que o planejamento desempenha ainda um papel vital na determinação dos roteiros, isto é, na determinação das máquinas assinaladas para processo e no tempo de entrada para processo.

2.1.1. Controle da Operação

A nível de planejamento, vários aspectos operacionais do sistema não podem ser capturados. Entretanto, durante a fase de execução, tais aspectos poderão possuir impacto significativo sobre o desempenho do sistema, como por exemplo aspectos relacionados às decisões que devem ser tomadas em tempo real, tais como o momento de liberação de uma peça no sistema, o tempo de início de uma operação ou de despacho de uma peça para ser transportada. Como lembra Ansoff [1977], as decisões operacionais tendem a ser aquelas que absorvem a maior parte da energia e da atenção de uma empresa, pois são estas que em última análise maximizam a eficiência do processo de conversão de recursos da empresa, ou, em linguagem mais convencional, procuram maximizar a rentabilidade das operações correntes. A complexidade da decisão e a necessidade de produzi-la em um curto intervalo de tempo obrigam o decisor a negligenciar muitos aspectos importantes, influenciando assim a qualidade do resultado.

A tentativa de solução através da modelagem matemática de problemas de controle operacional é uma solução que envolve um grande número de variáveis inteiras. Isto tende a dificultar bastante sua aplicação. Mesmo para problemas que envolvem sistemas com um único centro de usinagem, como o proposto e tratado por Raman e outros [1989], o método ainda é de difícil aplicação e solução, a menos que uma série de restrições seja previamente feita. Raman acredita que derive deste fato o grau de importância dado as decisões tomadas a nível de planejamento,

Revisão Bibliográfica

pois estas devem ser compulsoriamente encaminhadas antes das decisões oriundas do controle operacional, a fim de simplificá-las, isto é, a fim de torná-las exequíveis.

A seguir, revisa-se alguns dos problemas relacionados às decisões operacionais, tais como despacho de peças e veículos.

2.1.1.1. Despacho de Peças

O despacho de peças resume-se a dois problemas: o do gerenciamento ou ordenação da entrada de peças no sistema e o da ordenação ou priorização de peças que encontram-se nos *buffers* das máquinas.

Os problemas relativos a ordem de entrada das peças no sistema foram bem tratados em um estudo no qual Carrie e Petsopoulos [1985] utilizaram-se de um modelo de simulação. Neste estudo, sete diferentes regras de gerenciamento da entrada das peças foram testadas. Um experimento foi projetado para analisar a significância destas regras para a performance do sistema. Os resultados, embora significativos, foram dependentes dos tipos de peças, uma vez que as regras utilizadas estavam baseadas nas cargas de trabalho de cada tipo de peça.

Outros autores citam trabalhos de investigações sobre a importância das regras para entrada das peças.[ARBEL, 1984; MONTAZERI, 1990]. Seguindo a mesma linha investigatória, Maheshwari [1992] realizou um trabalho através de exaustivos experimentos utilizando várias regras de despacho, de forma individual ou combinada com regras de seqüenciamento de peças em filas, de despacho de veículos e de parâmetros operacionais. Suas conclusões, demonstram a importância destas regras sobre o desempenho operacional.

Revisão Bibliográfica

Com relação ao segundo tipo de problema associado ao despacho de peças, isto é, a ordenação ou priorização de peças que se encontram em filas ou em *buffers* de máquinas, o número de trabalhos na literatura é bastante amplo. Alguns destes trabalhos merecem atenção. Nof e outros [1979] reportam em uma das primeiras pesquisas voltada a células automatizadas de manufatura, a significância desta ordenação. Quatro diferentes regras foram utilizadas: FIFO, Menor Tempo de Processamento Restante (MTPR), Menor Tempo de Processamento (MTP) e Maior Tempo de Processamento. Os resultados apontaram para a significância das regras mas, ao mesmo tempo, citam que esta depende do *mix*, do tamanho dos lotes e da participação relativa de cada tipo de peça.

Stecke e Solberg [1981] modelaram um SFM da Caterpillar em Peoria, Illinois, EUA. O sistema era composto de nove máquinas, tendo um número de *pallets* e fixadores restritos. Dezesesseis regras de ordenação em conjunto com seis políticas de carregamento foram testadas. A combinação Menor Taxa de Produção (MTxP) para carregamento e Tempo Total de Processamento obteve o melhor desempenho. Os autores no entanto citam que este desempenho depende extraordinariamente dos parâmetros do sistema.

Com a hipótese de que uma combinação de regras obtém melhores resultados em um ambiente tipo *job-shop* do que uma única regra, Wu e Wysk [1989] desenvolveram um modelo de simulação para verificar qual combinação obteria os melhores resultados sobre um determinado sistema. O experimento foi realizado considerando diversos períodos de tempo, aos quais os autores denominaram “janelas de simulação”. Ao final de cada período, os resultados da simulação eram examinados e a melhor regra, com base no estado do sistema, era então escolhida para ser utilizada no período seguinte.

Revisão Bibliográfica

Em outro trabalho cujas decisões são tomadas com base no estado do sistema, Zeestraten [1990], desenvolveu um procedimento para a programação de SFM. O estado do sistema é baseado em um banco de dados , que é constantemente atualizado. A convergência do procedimento não foi discutida. O autor compara seu procedimento com dez diferentes regras de ordenação. Os resultados mostraram que seu procedimento obtivera os melhores resultados. Além disso é bastante consistente em termos da variância medida sobre o desempenho do sistema.

No mesmo trabalho investigatório já anteriormente citado, Maheshwari [1992], desenvolveu uma extensa pesquisa que resultou na elaboração de um modelo envolvendo tanto a fase de planejamento (com programação linear) como a fase de operação com simulação. Na fase de operação, diferentes regras para o controle operacional foram testadas, em combinação com quatro modelos de planejamento e diferentes parâmetros operacionais. Os resultados apontaram para diferentes combinações proporcionando melhor desempenho do sistema. De maneira geral, as combinações que obtiveram os melhores resultados envolveram a regra MTP para o controle das filas, sendo esta considerada a melhor regra quando se tratava de despacho de peças em filas.

Gupta e outros [1990] e Rachamadagu e Steckle [1988] desenvolveram e aplicaram uma série de regras de despacho de peças tanto para sistemas convencionais quanto para sistemas avançados de manufatura. Algumas destas regras foram aproveitadas por Gupta [1993] que concentrou seus estudos sobre quatro regras, a saber: SPT ou MTP, EDD (Earliest Due Date), CRATIO (Critical Ratio) e WINQ (Work In Next Queue). Verificou-se neste trabalho que SPT era uma regra dominante

Revisão Bibliográfica

para a maiorias das variáveis. Surpreendentemente, ainda segundo eles, as regras simples SPT e EDD superaram em desempenho as regras por eles chamadas de globais, isto é, regras que avaliam o sistema como um todo, como eram as regras CRATIO e WINQ. Acreditou-se que isto se devia ao tamanho médio das filas (entre 2 e 4). Por serem relativamente pequenos, a vantagem de aplicação de regras globais desaparecia.

Outros trabalhos que podem ser citados nesta área: [PANWALKER e ISKANDER, 1977], [DAR-EL e WYSK, 1982], [BLACKSTONE e outros, 1982], [WU, 1987] e [PHILIPOOM e FRY, 1990].

2.1.1.2 Despacho de Veículos

Em um sistema de produção operando sob encomenda, o problema de transporte se restringe a escolha do transporte e a determinação da peça a ser transportada. O transporte destas peças entre os diversas máquinas, ao final de cada uma das operações, é que gera o problema do despacho de veículos. As decisões tomadas sobre este problema, possuem considerável impacto sobre o desempenho de sistemas integrados de manufatura, como pode ser verificado nos trabalhos citados a seguir.

Em 1984, Egbelu e Tanchoco [1984], classificaram as tarefas de despacho de veículos em duas categorias. A primeira envolvia a seleção de um veículo vazio, por uma peça, quando do término de uma operação, havendo a necessidade de transporte. A segunda categoria, envolvia a seleção de uma peça para transporte, entre as várias que estejam esperando por transporte.

Em um procedimento sugerido por Raman e outros [1989] o transporte das peças era considerado como uma operação e o tempo de transporte como o tempo de uma operação. Assim, as

Revisão Bibliográfica

decisões de planejamento envolviam, simplificadaamente apenas operações, no seu sentido mais amplo. As prioridades eram estabelecidas de acordo com algumas regras baseadas no cumprimento de prazos de entrega. Os resultados apontaram alguns conflitos no uso dos veículos.

A análise de regras e dos procedimentos existentes para o despacho de veículos foi elaborado por Bozer [1989]. Ele ressalta a necessidade de um método consistente para a avaliação do desempenho do sistema, a fim de se poder verificar a significância de cada uma das regras, individualmente ou combinadas. A maioria dos testes foram realizados sob condições específicas, não podendo, portanto serem generalizadas. Importante neste trabalho, no entanto, foi esclarecer o fato de que regras para despacho de veículos não devem ser testadas independentemente de outras decisões relacionadas com o planejamento e controle de operações, tais como despacho de peças, ordenamento de peças ou alocação de máquinas.

Maheshwari [1992] também incluiu análises sobre a influência das regras de despacho de veículos sobre o desempenho. Suas conclusões apontam para uma regra baseada no Mínima Carga de Trabalho na Fila da Próxima Máquina (MCTPM), isto é, a peça selecionada será aquela cuja próxima fila de entrada de peças esperando para processo apresentar a menor carga de trabalho.

Outro trabalho retrata também a importância das regras de despacho de veículos. L. Zhuang e Hindi [1993], avaliaram um sistema de produção adotando o *lead time* das peças como medida o desempenho. Usando um modelo baseado em redes de filas, estudaram as interrelações entre regras de despacho e flexibilidade de rotas dentro de SFM e seu impacto sobre o desempenho operacional do sistema.

2.1.2. Impacto dos Parâmetros do Sistema

No planejamento é difícil formular as restrições impostas pelo sistema com base nos recursos logísticos e nas demais características tecnológicas do sistema. Estes recursos e características podem incluir o número de transportadores, o número de *pallets*, o *layout* do sistema ou o tamanho dos *buffers*. Estes parâmetros, possuem, definitivamente impacto sobre as características operacionais dos sistemas produtivos, como podem ser comprovados pelos seguintes trabalhos.

Em um estudo sobre sistemas com tamanhos limitados de *buffers*, Co e Jaw [1988], não encontraram muitas evidências sobre a importância das regras de ordenação de peças quando os *buffers* eram pequenos. Isto é, a utilização de diferentes regras de ordenação não produzia impacto significativo sobre o desempenho do sistema quando o número de peças a serem ordenadas nos *buffers* era limitado. No entanto, na medida em que crescem os tamanhos de *buffers*, crescem os estoques intermediários e a importância de seu gerenciamento sobre o desempenho dos sistemas. Este impacto foi verificado pelos autores quando do aumento dos *buffers* e do número de *pallets*.

Outros trabalhos a destacar são os de Schriber e Steckle [1988] e o de Maheshwari [1992]. Estes demonstram o impacto da variação dos parâmetros secundários do sistema sobre a taxa de utilização das máquinas e transportadores, e principalmente sobre o desempenho operacional do sistema. Os seus resultados demonstraram, por exemplo que um aumento no tamanho dos *buffers* ou no número de transportadores nem sempre melhoram a performance. Em alguns casos, o desempenho até decai em função de congestionamentos no sistema.

Freitas [1994] através da aplicação de um modelo utilizando sistemas especialistas e simulação, observou resultados

semelhantes àqueles obtidos por Maheshwari [1992] e Gupta [1992]. Contudo seus resultados foram melhores que os apresentados por Maheshwari, entre 6 e 22% medidos em relação aos tempos de ciclo de fabricação e entre 3 e 14% para tempo médio de fluxo. Freitas acredita que este ganho é proveniente da desconsideração, naqueles trabalhos, da dinâmica dos sistemas durante a fase de planejamento.

2.2. Aplicações de Inteligência Artificial para Otimização de Scheduling

Nos últimos anos, diversos trabalhos têm sido desenvolvidos, procurando utilizar técnicas de inteligência artificial (AI) em problemas de *scheduling*. Estes trabalhos procuram demonstrar que apesar da diversas alternativas disponíveis para execução do *scheduling*, é possível identificar padrões, acrescentar parâmetros difusos à avaliação, aplicar modelos de tomada de decisão e acoplar sistemas especialistas ou algoritmos otimizadores, separadamente ou conjuntamente, a fim de se obter um sistema autônomo de determinação de *scheduling*. A seguir são descritos alguns trabalhos relacionados ao uso de técnicas de AI em problemas de *scheduling*.

Com relação ao uso de sistemas especialistas, Freitas [1994], no mesmo trabalho já citado, propôs a utilização conjunta dos mesmos com simulação de processos para determinação de regras operacionais, através do desenvolvimento de um protótipo. Como resultado, verificou o incremento de desempenho de seu sistema em relação à mesma situação sem a existência desta ferramenta, o que segundo ele comprova sua utilidade.

A aplicação de redes neurais também tem crescido, conforme pode-se observar nos trabalhos de Arizono e outros [1992], que propuseram um sistema de minimização do tempo total de fluxo através de um sistema de máquina Gaussiana (modelo estocástico de rede neuronal). Concluíram que com este tipo de abordagem,

Revisão Bibliográfica

oferecia-se a possibilidade de solução de vários tipos de problemas de *scheduling*, sem que fosse alterada a dinâmica do sistema. Contudo vários problemas ainda continuavam presentes.

Outro trabalho digno de citação é o de Cho e Wysk [1993], que propuseram um modelo de gerador de *scheduling*, que pode ser visto na figura 2.1. Neste gerador, um sistema controlador de fábrica deveria repassar o planejamento da produção, feito por um sistema do tipo MRP (Material Requirement Planning), a um controlador de chão de fábrica, bem como os objetivos gerenciais determinados e as condições operacionais. Este controlador era encarregado de gerar o *scheduling*, assim como era função do controlador solicitar informações sobre o sistema. Em cada estação de trabalho haveria um controlador encarregado de gerar *scheduling* para as máquinas da estação em questão. No nível mais baixo estaria um controlador de máquina que executaria então a operação determinada pelos sistemas superiores. O sistema era composto de um pré-processador, que era encarregado de fornecer os parâmetros necessários para duas redes neurais, e para um simulador que se encarregaria de inferir sobre a solução apresentada. Como conclusão, observou a robustez do sistema proposto e verificaram que a “janela de simulação” utilizada possuía grande efeito sobre o desempenho do sistema. O resultado obtido foi significativamente superior a utilização de apenas uma estratégia baseada no critério de desempenho.

Revisão Bibliográfica

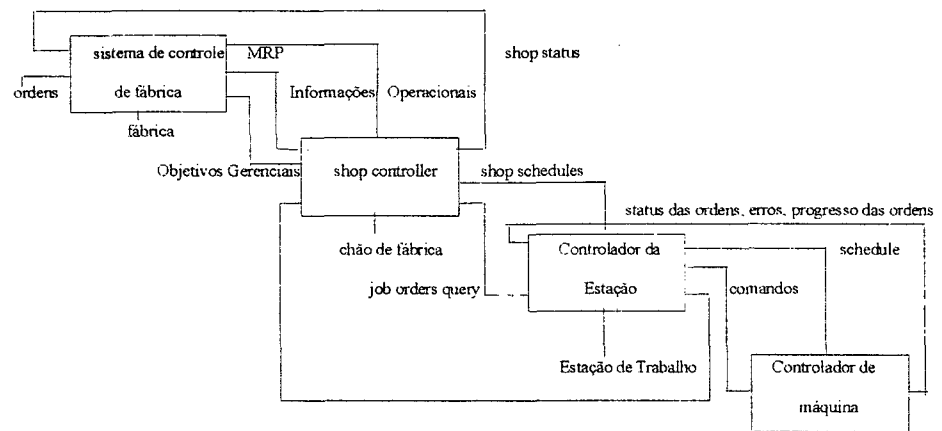


Figura 2.1. Fluxo de informações do modelo proposto por Cho e Wysk [1993].
Observa-se quatro níveis de decisão (sistema de controle de fábrica, um controlador de chão de fábrica, um controlador de estação e um controlador de máquina).

Ainda trabalhando com redes neurais, Watanabe [1993] desenvolveu um sistema para geração de *scheduling* para empresas que operam como sistemas *job-shop* utilizando as mesmas para avaliação das folgas (tempo de entrega - tempo atual). Este trabalho combinou a utilização de heurísticas para determinação de prioridades com a simulação de processos. Foram levados em consideração parâmetros relacionados ao número de operações da tarefa, data de entrega, tempo médio de operação, número de tarefas e número de máquinas. Watanabe concluiu que houvera melhora no desempenho do sistema, mas que os problemas de convergência ainda não teriam sido completamente resolvidos.

A aplicação de sistemas difusos também tem aparecido na literatura, conforme pode ser comprovado pelo trabalho de Grabot e Genestel [1994]. Neste trabalho, propuseram o uso de lógica difusa relacionada a aplicação de regras de seqüenciamento. A razão para isto, segundo os autores, era que quando houvesse variação dos parâmetros relacionados ao processo, haveria necessidade de alteração dos critérios de

Revisão Bibliográfica

aplicação das regras. Como conclusão, observou-se a possibilidade de se utilizar regras mais versáteis, isto é, regras que não fossem tão especializadas. Ainda, chegou-se a conclusão que esta abordagem poderia ser facilmente agregada a sistemas computacionais.

A utilização de algoritmos genéticos, apesar de ainda recente, começa a aparecer na literatura relacionada a engenharia industrial, como pode ser exemplificada pelo trabalho de Gauthier [1994]. Neste trabalho, onde um sistema de produção do tipo *flow-shop* é analisado, ferramentas como sistemas especialistas, lógica difusa e algoritmos genéticos são utilizadas. Comprovou-se através da aplicação em trinta problemas de produção, variando os tipos de peças e números de máquinas, que os resultados obtidos eram efetivamente melhores com respeito ao tamanho do lote intermediário, atraso médio e tempo total de produção, nesta ordem de importância.

2.3. Sumário

Neste capítulo abordou-se a literatura relevante aos propósitos da pesquisa. Primeiramente foi enfatizado o problema de planejamento e operação dos sistemas produtivos. Oportunamente revisou-se a aplicação de técnicas de inteligência artificial aos problemas de *scheduling*, particularmente ferramentas como sistemas especialistas, algoritmos genéticos, lógica difusa e redes neurais.

Por serem todos estes tópicos importantes para o desenvolvimento do gerador de *scheduling*, procurou-se revisar de forma mais profunda, subdividindo em seus principais segmentos, a fim de melhor compreender e apresentar o problema.

PRINCIPAIS FERRAMENTAS UTILIZADAS

Neste Capítulo serão analisadas as principais ferramentas utilizadas no modelo proposto para geração de *scheduling*. Além de ferramentas de AI (sistemas difusos e algoritmos genéticos), foi aplicada a simulação de processos visual interativa para verificação do funcionamento do mesmo em uma aplicação escolhida.

3.1. Simulação

Simulação pode ser definida como sendo “o desenvolvimento de um modelo lógico que reproduza a realidade, a fim de avaliar o comportamento e o desempenho de sistemas sob as mais variadas condições” [PEGDEN e outros, 1990]. Aqui considera-se simulação tanto a fase de modelagem do sistema quanto a fase de experimentação, bem como as metodologias aplicadas na descrição do comportamento dos sistemas; e as teorias ou hipóteses usadas na construção do modelo; sem deixar de citar o uso do sistema para previsão de comportamentos futuros, isto é, para avaliar os efeitos produzidos por alterações nos modelos. Em termos de seu uso, Pegden acredita que o uso de modelos de simulação é necessário quando se pretende apreender alguma coisa a respeito de alguns sistemas reais, e não se pode observar ou fazer experimentos diretamente, seja por ainda não existirem ou pela dificuldade em lidar com os mesmos em situação real. Lembra ainda que um modelo cuidadosamente concebido pode eliminar grande parte da complexidade inicialmente imaginada, deixando apenas características que o analista considere como importantes.

Principais Ferramentas Utilizadas

Historicamente, devido ao aumento da competitividade, as indústrias tem sido levadas cada vez mais a promoverem caros processos de automação fabril. Isto implica geralmente no reexame e na redefinição das políticas e dos procedimentos existentes. Infelizmente, até mesmo o mais cuidadoso planejamento não é capaz de evitar a ocorrência de falhas, tais como: erros na determinação das capacidades de máquinas; insuficiência de espaços nos *buffers*; bloqueios de transportes, como por exemplo pontes rolantes, entre outros. Neste contexto, a simulação ganha importância como ferramenta de análise que possibilita o estudo *a priori*, evitando a ocorrência destes tipos de erros.

As ferramentas de simulação podem ser empregadas tanto nas fases de planejamento, quanto nas fases de projeto ou de operação de sistemas de produção, agindo sobre três áreas, a saber:

- a) *Hard-system*: concerne ao número e aos tipos dos equipamentos, a configuração do *layout* e as peças a serem processadas pelo sistema;
- b) *Soft-system*: refere-se ao planejamento, seqüenciamento das peças, *scheduling*, ferramentas, trabalhadores;
- c) Controle em tempo-real: controle de fluxo envolvendo contingências como falha de ferramenta, máquinas ou paradas, etc.

Isto implica no uso futuro da simulação em três modos distintos:

- a) como ferramenta de projeto e análise de *layouts*, de compra de máquinas, de avaliação de políticas alternativas, etc. Atualmente,

Principais Ferramentas Utilizadas

estas são as funções da simulação mais frequentemente utilizadas;

b) como ferramenta de *scheduling*, particularmente em se tratando de sistemas automatizados. Este uso permite ao tomador de decisão explorar, planejar e avaliar mudanças no *scheduling* a fim de obter a otimização do mesmo.;

c) como parte de sistemas em tempo real, ou de controles *on-line*. Tal sistema deverá ser periodicamente ativado, ler as condições correntes, definindo o *scheduling*, com ou sem a intervenção humana.

3.1.1. Vantagens e Desvantagens do Uso da Simulação

Devido ao fato de ser facilmente compreendido, um modelo de simulação é frequentemente mais simples de ser justificado do que alguns modelos analíticos. Além disto um modelo de simulação costuma ter mais credibilidade, uma vez que pode ser comparado com o sistema real, ou porque requer pouca simplificação, capturando mais as características reais do sistema.

Todos os modelos de simulação são chamados de modelos de entrada e saída, isto é, eles produzem uma saída dada as condições das entradas. Os modelos de simulação não podem gerar por si mesmo uma solução ótima como é o caso dos sistemas analíticos. Eles servem apenas como ferramenta de análise do comportamento do sistema sob uma determinada condição.

Alguns dos benefícios do uso da simulação são listados abaixo [PEGDEN e outros, 1990]:

Principais Ferramentas Utilizadas

- a) novas políticas, procedimentos operacionais, regras de decisão, estruturas organizacionais, fluxos de informação, etc., podem ser explorados sem que provocar distúrbios nos processos em uso;
- b) novos projetos de *layout*, sistemas de transporte, máquinas e equipamentos, softwares, podem ser testados antes de sua implantação, avaliando assim a necessidade de compra ou modificação;
- c) hipóteses sobre como e por que certos fenômenos ocorrem podem ser testados;
- d) o fator tempo pode ser controlado, isto é, pode ser expandido ou comprimido, permitindo aumentar ou diminuir a velocidade a fim de se estudar um fenômeno;
- e) permite a análise de quais variáveis são significativas para o desempenho do sistema e como estas variáveis se interagem;
- f) gargalos podem ser identificados;
- g) um trabalho de simulação pode ser comprovadamente importante para o entendimento de como o sistema realmente funciona;
- h) novas situações, onde há pouca informação ou conhecimento a respeito, podem ser manipuladas a fim de se prever eventos futuros, isto é, a simulação é uma poderosa ferramenta para responder questões do tipo “o que acontecerá se ...”.

A despeito das várias vantagens da utilização da simulação, deve-se ressaltar algumas restrições ou dificuldades na implantação de um modelo de simulação. As principais são [PEGDEN e outros, 1990]:

Principais Ferramentas Utilizadas

- a) necessidade de treinamento, uma vez que a qualidade da análise depende da qualidade do modelo, e portanto da habilidade do modelador;
- b) algumas vezes os resultados da simulação podem ser de difícil interpretação. Isto é devido ao fato da simulação tentar capturar a aleatoriedade de um sistema real, levando a dificuldade de identificação se um evento ocorreu devido a aleatoriedade ou a interações de elementos do sistema;
- c) análises feitas através do uso de simuladores podem ser demoradas e caras, podendo até mesmo inviabilizar seu uso.

3.1.2 O Processo de Simulação

O desenvolvimento de um modelo de simulação é um processo estruturado, onde o modelador do sistema deverá seguir alguns passos que estão listados abaixo [PEGDEN e outros, 1990]:

- a) definição do problema: deve-se definir claramente os objetivos do estudo, isto é, determinar por que do estudo e o que é esperado como resultado;
- b) planejamento do projeto: define-se aqui os recursos a serem utilizados;
- c) definição do sistema: determinação dos limites e restrições na definição do sistema ou processos;
- d) formulação do modelo conceptual: desenvolvimento de um modelo preliminar para definição dos componentes, variáveis descritivas, interações e lógicas do sistema;

Principais Ferramentas Utilizadas

- e) projeto experimental preliminar: seleção dos fatores e dos níveis a serem utilizados, além da escolha dos fatores a serem investigados, isto é, quais dados devem ser considerados no modelo, em qual forma e extensão;
- f) preparação dos dados de entrada: identificação e coleta dos dados necessários para a modelagem;
- g) modelagem: formulação do modelo em uma linguagem de simulação apropriada;
- h) verificação e validação: feito através da confirmação do funcionamento do modelo, isto é, se o modelo funciona como o modelador previu e se os dados de saídas são plausíveis;
- i) projeto final do experimento: projeta-se o experimento a fim de que se produza as informações desejadas e como cada rodada de simulação irá ser feita;
- j) experimentação: execução da simulação para gerar os dados necessários e desenvolver a análise de sensibilidade;
- l) análise e interpretação: produz-se inferências a partir dos dados produzidos pela simulação.

3.1.3. Aplicação de Simulação de Processos em Scheduling

Vários trabalhos utilizam a simulação de processos para comprovar o funcionamento de seus modelos propostos, bem como implementar a capacidade de tomada de decisão em tempo real, como pode ser verificado nos trabalhos descritos a seguir:

Lecomte e Dejax [1991] desenvolveram um protótipo de sistema de suporte a decisão (DSS) para problemas controle de sistemas produtivos do tipo *job-shop*. A simulação incorporava regras de

Principais Ferramentas Utilizadas

seqüenciamento, visando avaliá-la para geração de um novo *scheduling*.

Cho e Wysk [1993] também utilizaram um simulador no seu modelo proposto. O simulador era usado para selecionar a melhor regra de despacho com base nas regras recomendadas por uma rede neuronal. O simulador calculava a melhor de duas regras através de critérios de desempenho determinados.

Outra importante aplicação foi feita por Gupta e outros [1993]. Analisando problemas de planejamento operacional e *scheduling*, empregaram um detalhado modelo de simulação, a fim de gerar saídas para um analisador estatístico. Este modelo incorporava regras de liberação de peças para entrada no sistema, assim como regras de despacho de peças para fabricação.

Anderson e Nyirenda [1990] desenvolveram duas novas regras de seqüenciamento aplicáveis em problemas de minimização de atrasos. Estas regras foram comparadas com outras regras básicas através do uso de simulação de processos, demonstrando a efetividade das mesmas.

Outros trabalhos como o de Ding e outros [1990], Yang e Sum [1994], Bais e Nagarur [1994] e Freitas [1994] também utilizaram a simulação para a avaliação de seus modelos ou regras.

3.2. Sistemas Difusos

Sistemas difusos são conjuntos de associações difusas, ou de regras, que admitem graus de pertinência. Este tipo de sistema já está disponível em vários controladores comercialmente disponíveis, tais como: condicionadores de ar, transmissão de automóveis, máquinas de lavar, semáforos, entre outros.

Principais Ferramentas Utilizadas

No processamento, os sistemas difusos trabalham com inferências em paralelo, obtendo como resposta diferentes graduações para cada uma das regras colocadas, ao invés de estabelecer respostas bivalentes, como derivadas da estrutura lógica dual.

A aplicação de controladores para sistemas complexos exige a utilização de controladores adaptativos, a fim de possibilitar que o mesmo aprenda a manipular o sistema a partir de algumas regras básicas. Estes sistemas de controle adaptativos são sistemas que estimam a saída a partir de um conjunto de dados básicos, sem que seja necessário especificar matematicamente a função resposta [KOSKO, 1992]. Sucessivos refinamentos e iterações fazem com que o sistema aperfeiçoe seu desempenho ao longo do tempo.

Aplicou-se neste trabalho controladores adaptativos em sistemas de produção devido ao fato que nestes sistemas há geralmente a presença de várias alternativas possíveis de implementação, bem como há normalmente uma dificuldade de se prever com razoável precisão o comportamento do sistema ao longo do tempo, tornando assim difícil sua modelagem matemática. A opção por um controlador adaptativo difuso é resultado da facilidade em desenvolver este tipo de controlador dentro do simulador, ao mesmo tempo que permite com extrema facilidade sua conexão ao algoritmo otimizador, desenvolvido externamente.

3.2.1. Regras Difusas

O exemplo mais simples de uma regra difusa consiste em uma associação única, por exemplo, uma associação do tipo (PESADO, LONGO) para um controlador de tráfego. Em geral um sistema de regras, normalmente baseados em FAM (Fuzzy Associative Memories), consiste em um banco de diferentes associações. Cada associação corresponde a uma diferente matriz numérica, ou a uma diferente entrada em um banco de matrizes

Principais Ferramentas Utilizadas

lingüísticas. Estas matrizes são guardadas em separado, fazendo-se acesso a elas em paralelo, a fim de evitar a ocorrência de interferências.

Um exemplo de associação única é o caso do controlador de tráfego, acima citado. Aqui um par de conjuntos difusos (A, B) representa uma associação do tipo (LEVE, CURTO), isto é, para tráfego leve aplica-se um intervalo curto entre troca de sinais. Quantitativa-se o domínio da densidade de tráfego através de variáveis numéricas $(x_1, x_2, x_3, \dots, x_n)$, sendo os mesmos dados, por exemplo, em carros por minuto. Quantitativa-se a faixa de duração da luz verde por p variáveis $(y_1, y_2, y_3, \dots, y_p)$, podendo ser os mesmos dados em segundos. Os elementos x_i e y_i pertencem respectivamente aos conjuntos básicos $X = \{x_1, \dots, x_n\}$ e $Y = \{y_1, \dots, y_p\}$, que são os conjuntos de todos os valores numéricos possíveis de entrada e saída. Os conjuntos difusos A e B, definem as funções de pertinência m_A e m_B que fazem o mapeamento dos elementos x_i de X e y_j de Y para valores de pertinência no intervalo de $[0,1]$. Estes valores de pertinência indicam o quanto x_i pertence ao conjunto A e quanto y_j pertence ao conjunto B. Representa-se estas funções por: $m_A: X \rightarrow [0,1]$ e $m_B: Y \rightarrow [0,1]$. A e B são vetores numéricos dados por: $A = (a_1, \dots, a_n)$ e $B = (b_1, \dots, b_n)$, onde $a_i = m_A(x_i)$ e $b_j = m_B(y_j)$.

A Figura 3.1 ilustra três subconjuntos do universo de discurso X. Neste exemplo existem três subconjuntos difusos de entrada [A_1 (leve), A_2 (médio) e A_3 (pesado)] pertencentes ao conjunto A. Para cada valor de x pertencente ao universo de discurso X (densidade de tráfego), haverá três valores de pertinência (m_{A_1} , m_{A_2} , m_{A_3}) relacionados aos subconjuntos A_1 , A_2 e A_3 . Para um valor de x igual a 45, por exemplo, os valores das pertinências serão respectivamente (0,65; 0,45; 0). Estes valores indicam que existe uma maior afinidade entre este valor de x e o subconjunto

Principais Ferramentas Utilizadas

A_1 (leve), bem como a afinidade do subconjunto A_2 (médio) é maior que a a afinidade do valor x ao subconjunto A_3 (pesado).

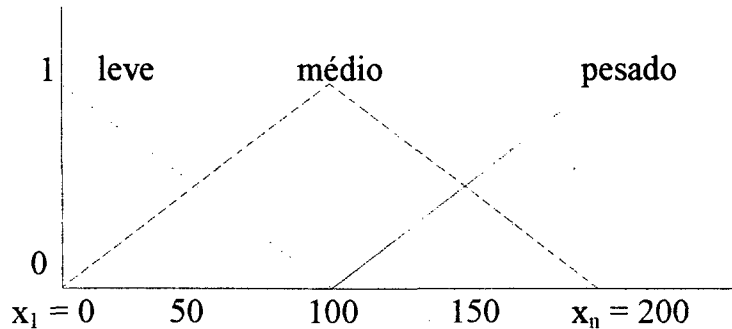


Figura 3.1. Três subconjuntos difusos do espaço X (densidade de tráfego).

3.2.2. Tomada de Decisão Multi-Atributo Difusa

Segundo Zimmermann [1992], o problema de tomada de decisão multi-atributo também pode ser tratado através da agregação de lógica difusa. Similarmente a um problema *crisp*, tem-se:

$X = \{x_i\}$, $i = 1, \dots, n$ é um conjunto finito de alternativas

$G = \{g_j\}$, $j = 1, \dots, m$ é um conjunto finito de objetivos, atributos ou critérios.

Pode-se priorizar as alternativas através de alguns métodos existentes. Um dos mais simples é o modelo estocástico de Kahne, que pode ser visto abaixo:

$$R_i = \sum_{j=1} w_j \cdot r_{ij} / \sum_{j=1} w_j \quad (\text{eq. 3.1})$$

onde,

r_{ij} = relação da i -ésima alternativa com o j -ésimo critério

w_j = importância relativa do critério j .

R_i = posição no Ranking da alternativa i

Para se obter a alternativa ótima maximiza-se R_i

Principais Ferramentas Utilizadas

Agregando-se a lógica difusa este modelo, como sugerido por Yager [1978], temos:

$$r_{ij} = \{(r_{ij}, mr_{ij}(r_{ij}))\} \quad (\text{eq. 3.2})$$

onde,

r_{ij} = relação da i-ésima alternativa com o j-ésimo critério

mr_{ij} = pertinência da relação r_{ij} com o objetivo, atributo ou critério.

A alternativa desejada é dada por:

$$X_{\text{opt}} = \text{MAX}_i \text{MIN}_j C_j, \text{ onde } C_j \text{ se refere ao } j\text{-ésimo critério.} \quad (\text{eq. 3.3})$$

Exemplo:

Sejam, os conjuntos

$X = \{x_i\}, \quad i = 1, 2, 3 \quad (\text{alternativas})$

$C = \{g_j\}, \quad j = 1, \dots, 4 \quad (\text{critérios})$

$w = \{w_j\}, \quad j = 1, \dots, 4 \quad (\text{pesos})$

dados por:

$C_1 = \{(x_1; 0,7), (x_2; 0,5), (x_3; 0,4)\}$

$C_2 = \{(x_1; 0,3), (x_2; 0,8), (x_3; 0,6)\}$

$C_3 = \{(x_1; 0,2), (x_2; 0,3), (x_3; 0,8)\}$

$C_4 = \{(x_1; 0,5), (x_2; 0,7), (x_3; 0,2)\}$

$W_j = \{2,32; 1,2; 0,32; 0,16\}$

Principais Ferramentas Utilizadas

Aplicando os pesos exponencialmente, como sugerido por Yager [1978], isto é elevando-se os graus de pertinência em termos de w_j , temos:

$$C_1 = \{(x_1; 0,44), (x_2; 0,20), (x_3; 0,12)\}$$

$$C_2 = \{(x_1; 0,24), (x_2; 0,76), (x_3; 0,54)\}$$

$$C_3 = \{(x_1; 0,50), (x_2; 0,68), (x_3; 0,93)\}$$

$$C_4 = \{(x_1; 0,90), (x_2; 0,69), (x_3; 0,77)\}$$

Aplicando o método “Máximo dos Mínimos” para selecionar a alternativa desejada,

$$X_{opt} = \text{MAX}_i \text{MIN}_j C_j$$

onde,

$$\text{MIN}_j C_j = \{(x_1; 0,24), (x_2; 0,2), (x_3; 0,12)\} \quad (\text{eq. 3.4})$$

logo,

$$X_{opt} = \text{MAX}_i \text{MIN}_j C_j = \{(x_1; 0,24)\}$$

ou seja, a alternativa pretendida é a x_1 , levando-se em consideração todos os critérios estabelecidos.

3.3. Algoritmos Genéticos

Algoritmos genéticos (GAs) são algoritmos de pesquisa baseados no mecanismo de genética e na seleção natural, combinando a capacidade de sobrevivência de elementos mais adaptados com a possibilidade de criação de novos elementos [GOLDBERG, 1989]. Inicialmente desenvolvidos por John Holland e sua equipe da University of Michigan com o objetivo de explicar rigorosamente o mecanismo de funcionamento do processo adaptativo dos sistemas naturais e projetar softwares que

Principais Ferramentas Utilizadas

mantivessem esta característica, possibilitou a condução de importantes descobertas na ciência dos sistemas naturais e artificiais. No Anexo III são descritos os principais termos genéticos utilizados pelos GAs.

O tema central da pesquisa foi a procura por soluções robustas, e o balanceamento entre eficiência e eficácia relativos a sobrevivência em vários tipos de ambiente. Tal necessidade se deve ao fato de que se os sistemas forem mais robustos, os custos oriundos de reprojeto poderiam ser minimizados ou eliminados.

Segundo Beasley e outros [1993], os GAs diferem dos métodos tradicionais por:

- a) trabalhar com uma codificação do conjunto de parâmetros, e não propriamente com o parâmetro;
- b) pesquisar em uma população de pontos e não em um único ponto;
- c) usar função objetivo e não funções derivadas ou auxiliares;
- d) utilizar regras probabilísticas e não determinísticas.

Ainda segundo Beasley e outros [1993] e Goldberg [1989], o mecanismo de funcionamento dos GAs podem ser considerados extremamente simples, não envolvendo nada além de cópias e permutações de parte dos cromossomos (na nomenclatura dos GAs, *strings*). GAs simples que produzem bons resultados em vários problemas práticos são compostos de três operações fundamentais:

Principais Ferramentas Utilizadas

- a) reprodução, que é o processo em que *strings* individuais são copiadas de acordo com os valores das funções objetivos (para biólogos, função de *fitness*), isto é, as funções com maiores valores de *fitness* possuem maior probabilidade de contribuir com um ou mais descendentes na próxima geração. Este operador é uma versão artificial da seleção natural de Darwin. A maneira mais simples de criar este operador é através de um sorteio onde a probabilidade de ocorrer um evento “x” é diretamente proporcional ao valor dos *fitness*;
- b) *crossover*, fenômeno que ocorre após a reprodução dos elementos gerados, através de permutação de parte das *strings* de indivíduos escolhidos aleatoriamente. Há, portanto, a necessidade da determinação da posição onde serão partidas as *strings* selecionadas.

Exemplo: ocorrência de *crossover* entre dois indivíduos de uma população na posição 4 das *strings*. Observa-se a representação da *string* em números binários (esta representação se dará em todo o trabalho).

	<i>string</i> antes do <i>crossover</i>	<i>string</i> após <i>crossover</i>
Indivíduo 1	11110000	11111111
Indivíduo 2	00001111	00000000
posição escolhida = 4		

- c) mutação, em GAs é uma alteração aleatória e ocasional do valor de uma posição da *string*, visando garantir que não haja perda de características importantes, isto é, a mutação proporciona o aparecimento ou reaparecimento de características não existente na população atual.

Principais Ferramentas UtilizadasBiblioteca Universitária
UFSC

Exemplo:

	<i>string</i> antes da mutação	<i>string</i> após mutação
Indivíduo n	11110000	10110000
mutação na posição 2 da <i>string</i>		

**3.3.1. Algumas
Aplicações dos
GAs**

Os GAs possuem aplicações em várias áreas de conhecimento, como por exemplo na matemática, medicina, engenharia, ou ciência política.

Surgida da simulação computacional de sistemas genéticos por biólogos. [BARRICELLI, 1957, 1962; FRASER, 1960, 1962; MARTIN E COCKERHAM, 1960], os GAs acabaram por desenvolver aplicações em outras áreas. Embora os estudos fossem inicialmente conduzidos para análise de fenômenos naturais, estes, em especial o trabalho de Fraser, não estavam distantes da moderna noção dos GAs. Neste trabalho, Fraser procurou demonstrar a interação entre diversos parâmetros, simulando a evolução de gerações de cromossomos. Apesar disto, Holland foi o primeiro a reconhecer a aplicação de operadores genéticos em sistemas artificiais adaptativos. Os objetivos propostos eram amplos, conforme pode ser observado na transcrição abaixo [HOLLAND, 1962];

“O estudo sobre a adaptação envolve a pesquisa tanto do sistema adaptativo como do ambiente. Em termos gerais, é um estudo de como os sistemas podem gerar procedimentos que os habilite eficientemente no ajuste ao ambiente ...”

O objetivo de Holland não era, simplesmente, desenvolver outro método qualquer de otimização, mas sim criar uma abordagem teórica, assim como procedimentos para desenvolvimento de

Principais Ferramentas Utilizadas

programas genéricos e máquinas com capacidade ilimitada de adaptação a ambientes arbitrários.

O termo Algoritmos Genéticos foi primeiramente utilizado por Bagley [1967], assim como foi dele a primeira aplicação concreta dos GAs. O objeto do estudo foi um jogo de tabuleiro conhecido como “Hexapawn”.

Outros trabalhos a ser citados são os de: Cavicchio [1970], que aplicou os GAs em um problema de reconhecimento de padrões; Weinberg [1970] na simulação de células vivas; Hollstien [1971], sendo o primeiro a aplicar os GAs em problemas de otimização de funções matemáticas.

Atualmente os GAs tem sido utilizados em vários tipos de aplicações no campo da engenharia. Alguns exemplos são: Otimização de sistemas de gasodutos [GOLDBERG, 1983], layout de circuitos VLSI [DAVI, 1985] e a otimização do peso do braço do trem de pouso de aeronaves [MINGA 1986].

Em problemas de engenharia industrial, onde as aplicações ainda são escassas, aparecem com destaque os trabalhos de Davis [1992], que trabalhou com a alocação de espaços em Bins, isto é, utilizou os GAs para calcular a melhor combinação de colocação de objetos em espaços limitados.

Com relação aos trabalhos sobre *scheduling*, alguns já foram publicados. Além do trabalho de Gauthier [1994], já citado no item 2.2, os trabalhos de Davis [1985], Syswerda [1991], Whitley e Starkweather [1989], Bagchi e outros [1991] se concentraram no problema de determinação do melhor alocação de recursos a tarefas, visando a diminuição do tempo de *leadtime* e a maximização da utilização dos recursos.

Principais Ferramentas Utilizadas

Cleveland e Smith [1984] concentraram em problemas relacionados a liberação de ordens de produção em plantas do tipo *job-shop*. Aplicaram o mesmo método usado em problemas da determinação de rotas para “caixeiros viajantes”. Compararam os resultados com as regras SPT, EDD e LST e constataram a aplicabilidade deste método

Preocupando-se com o problema da programação da produção, Husband, Mill e Warrington [1992], propuseram um método de solução através da evolução das ordens de produção geradas. Constataram por fim a evolução da performance das ordens geradas.

3.3.2. Comparação da Terminologia da Genética e dos GAs

Apresenta-se neste item a comparação da terminologia utilizada pela genética e pelos GAs, pois alguns termos foram alterados na terminologia dos GAs. Por isso torna-se necessário explicitá-los a fim de proporcionar uma padronização dos termos utilizados neste trabalho. São dados pela tabela 3.1.

GENÉTICA	GAs
cromossomo	<i>string</i>
gene	característica
alelo	valor de uma característica
locus	posição na <i>string</i>
genótipo	conjunto de parâmetros
epistase	não linearidade

Tabela 3.1 Termos alterados pela teoria dos GAs em relação à genética.

Principais Ferramentas Utilizadas

3.4. Sumário

Neste capítulo abordou-se as teorias relevantes das principais ferramentas utilizadas no modelo proposto para geração de *scheduling*, que são a simulação, os algoritmos genéticos e a lógica difusa. Procurou-se através da determinação de um capítulo exclusivo para as mesmas, revisar mais profundamente os pontos considerados importantes para compreensão do problema e da abordagem utilizada, bem como fornecer subsídios a novas pesquisas.

O MODELO PROPOSTO PARA A GERAÇÃO DE SCHEDULING

Neste capítulo será mostrado o modelo proposto para geração de *scheduling*. Será dada especial atenção à descrição dos módulos componentes do modelo, isto é, ao modelo de simulação, aos controladores difusos, aos processadores centrais e ao algoritmo otimizador, bem como aos elementos principais da política operacional (regras utilizadas).

4.1. Razões para Desenvolvimento de Um Gerador de Scheduling

Profundas mudanças em uma ambiente cada vez mais competitivo, tendo como resultado um aumento da incerteza relacionada ao processo decisório empresarial, vem conduzindo as empresas a buscar tanto o aumento do desempenho, quanto o aumento da flexibilidade de seus sistemas produtivos. A fim de se alcançar os objetivos neste contexto, a administração passou a ter que gerenciar mais informações, mais restrições, bem como mais objetivos contraditórios. Vários sistemas auxiliares ao processo de tomada de decisão têm, ao longo dos anos, sido desenvolvidos, visando, sobretudo, tornar possível a manipulação destes diversos fatores, e fatalmente tenderão a ter seu uso compulsório nas empresas de maior complexidade e flexibilidade [LECOMTE e DEJAX, 1991].

Dos diversos fatores normalmente estudados na literatura sobre sistemas de produção, verifica-se que a atenção dos pesquisadores recai, freqüentemente, sobre o planejamento operacional e sobre os problemas de *scheduling*. Uma das principais razões para isto é que estes pontos implicam diretamente no desempenho geral do sistema, isto é, internamente influenciam na utilização dos recursos e externamente na resposta do sistema à variação da demanda. A literatura mais recente sobre

O Modelo Proposto Para Geração de Scheduling

OPS (*Operational Planning and Scheduling*) tem focalizado duas áreas principais: a) alocação de máquinas; b) despachos *on-line* [GUPTA e outros, 1991] e [RACHAMADUGU e STECKE, 1988].

Neste trabalho, não serão considerados problemas envolvendo alocação de máquinas. A razão para isto é que na aplicação proposta tal problema não existe.

Com relação aos problemas de despacho, métodos analíticos, tais como a programação inteira e a programação dinâmica, embora possam resolver este tipo de problema, só são aplicáveis na prática a problemas de tamanho limitado [CHO e WYSK, 1993]. Além disto, os métodos matemáticos existentes não podem ser usados para problemas de *scheduling* feitos *on-line* [BLACKSTONE e outros, 1982], [WU, 1987], [PHILLIPOOM e FRY, 1990]. Salienta-se, ainda, que, segundo Cho e Wysk [1993], o melhor caminho para solucionar problemas de *scheduling* em tempo real é o uso de estratégias dinâmicas de despachos, isto é, através do uso de controladores que fazem a seleção da regra a ser aplicada ao longo do tempo, conforme a condição do sistema.

Uma das abordagens atualmente mais utilizada e que permite a consideração de fatores como complexidade do sistema e a capacidade de geração de *scheduling on-line*, é através do desenvolvimento de um modelo de simulação que permita agregar estes pontos ao nível de detalhe pretendido. Isto será basicamente o que o sistema aqui desenvolvido se proporá a fazer.

O Modelo Proposto Para Geração de Scheduling

4.2. Elementos do Planejamento da Produção

Antes de propriamente iniciar a descrição do modelo proposto, é importante explicitar os principais elementos do planejamento da produção, uma vez que algumas destas características serão aproveitadas no desenvolvimento do modelo.

Na figura 4.1 estão ilustrados os principais elementos envolvidos no processo de planejamento da produção.

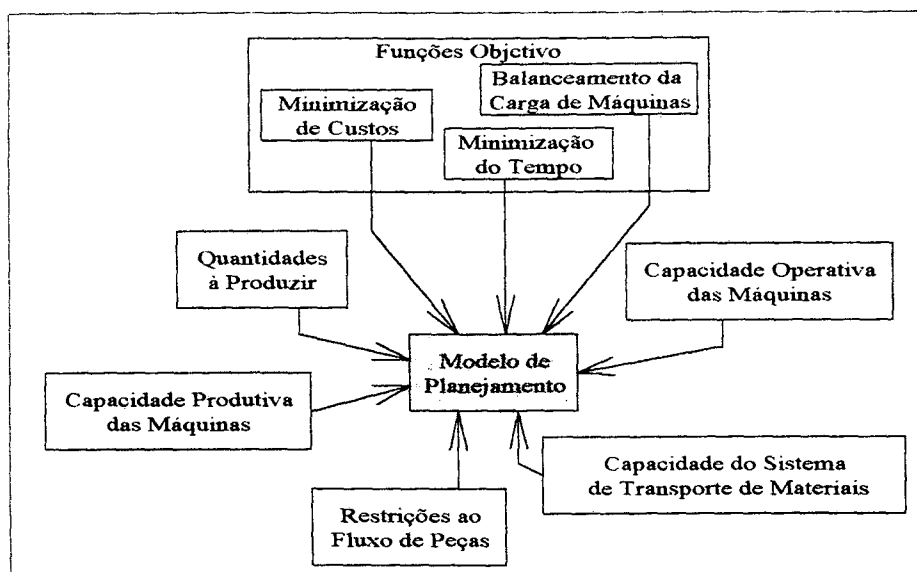


Figura. 4.1 - Elementos dos Modelos de Planejamento [FREITAS, 1994]

Em seu mais baixo nível de decisão, o planejamento requer do decisor, a definição da programação de todas as peças pertencentes ao lote de produção a ser manufaturado ao início de cada ciclo produtivo, que pode ser um turno, um dia, uma semana, ou outro período qualquer.

Esta decisão, envolve vários parâmetros de planejamento tais como as necessidades de produção, por tipo de peça, bem como, as capacidades produtiva e operacional de cada máquina do sistema. Ao lado destas variáveis, o decisor deve ainda considerar a capacidade do sistema de transporte de materiais, elemento de fundamental importância, como salienta Freitas [FREITAS, 1994].

O Modelo Proposto Para Geração de Scheduling

Paralelamente às necessidades de produção e às restrições do sistema, é necessário que o decisor especifique um objetivo a ser alcançado com o planejamento. Vários objetivos diferentes são possíveis tais como:

- Minimização do custo operacional;
- Minimização da soma dos tempos de operação e transporte;
- Minimização do máximo atraso ou atraso médio das peças para entrega;
- Balanceamento da carga de cada centro de usinagem, etc.

A solução destes problemas de planejamento da produção, passa, em geral pela formulação de modelos de programação linear inteira, como os apresentados em STECKE [1986] e KUSIAK [1985]. A grande dificuldade porém, não se encontra nesta fase, mas durante a implementação operacional das soluções obtidas.

4.3. Elementos do Controle Operacional

No item anterior apresentou-se alguns dos principais elementos presentes nas decisões relacionadas com o planejamento da produção. Entretanto, conforme visto no capítulo anterior, na etapa de execução muitas outras decisões de controle devem ser tomadas, as quais poderão afetar o planejamento e como consequência o próprio desempenho do sistema.

No nível de planejamento os modelos utilizados não incorporam muitas políticas operacionais com comprovada influência sobre o desempenho do sistema. Dentre as decisões operacionais de controle mais importantes tem-se, segundo Maheshwari, [1992]:

- Políticas para a liberação de novas peças ao sistema;
- Políticas para a priorização de peças no *buffer* das máquinas;
- Políticas para o despacho de veículos transportadores

O Modelo Proposto Para Geração de Scheduling

A figura 4.2 ilustra o controle operacional, sua dependência das políticas operacionais, dos recursos secundários do sistema e das decisões tomadas na fase de planejamento.

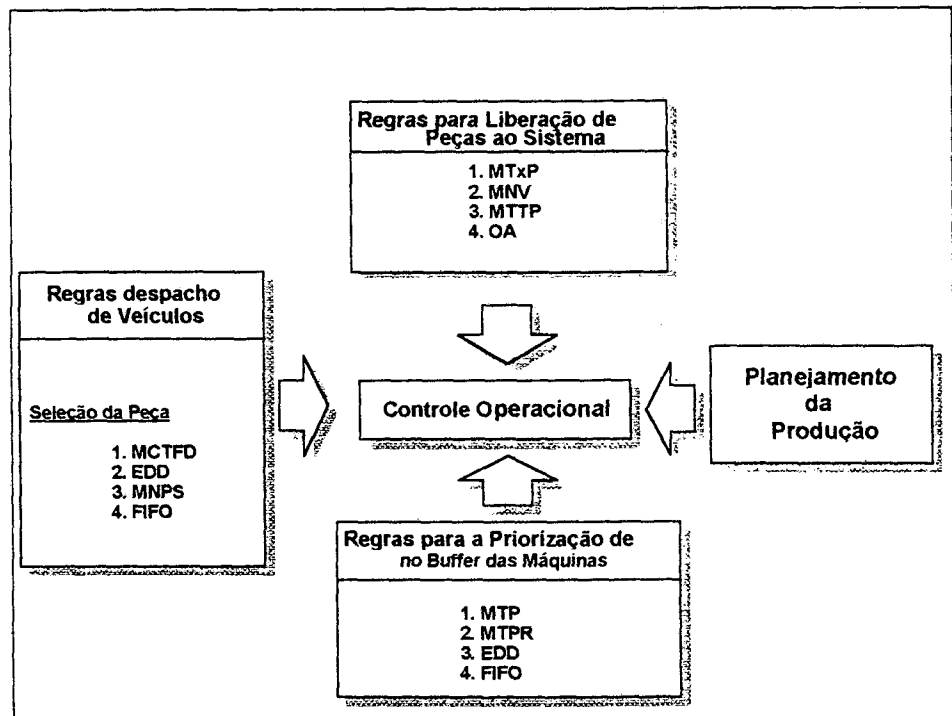


Figura 4.2. Elementos do Controle Operacional [FREITAS, 1994]

Nos subitens que se seguem apresenta-se uma descrição das políticas ou regras operacionais citadas acima

4.3.1. Regras para a Liberação de Novas Peças ao Sistema

Estas políticas ou regras, determinam, dentre as peças que estão aguardando para serem introduzidas no sistema, qual terá a maior a prioridade. Uma regra de prioridade deve ser definida para determinar qual dos diferentes tipos de peças que se encontram aguardando ser a escolhida. A literatura reporta vários trabalhos que tratam destas regras [NOF, 1979; CARRIE, 1985; WHITNEY, 1985; MONTAZERI, 1990; GUPTA, 1992].

O Modelo Proposto Para Geração de Scheduling

Tais regras são, em geral, dependentes de características das peças, tais como, tempos de processamento, número de operações, tempo de chegada ou do estado do sistema, que pode ser descrito, por exemplo, pela situação em que se encontram as máquinas que uma peça deve visitar ou pela taxa de produção do *mix* naquele momento. As regras mais utilizadas são:

- *Menor Taxa de Produção (MTxP)*: Esta regra depende do estado do sistema. É executado um acompanhamento, constante, da relação entre os tipos de peças já introduzidas no sistema, e o *mix* de produção desejado. É introduzida no sistema a peça com a menor relação. Esta regra procura manter o *mix* de produção desejado ao longo de todo o ciclo produtivo.
- *Menor Número de Visitas (MNV)*: Esta regra libera peças para o sistema, com base no número de visitas que uma peça tem que realizar para completar todas as suas operações, em uma ordem cíclica.
- *Menor Tempo Total de Processamento (MTTP)*: Esta regra libera peças para o sistema, com base no tempo total de processamento necessário para que um peça realize todas as suas operações, também em uma ordem cíclica.
- *Ordem Aleatória (OA)*: Neste caso todas as peças aguardando possuem a mesma probabilidade de ser escolhida. Em geral este tipo de regra é adotado como forma de comparação no desempenho de outras regras

Especificamente, neste trabalho, não serão utilizadas estas regras, uma vez que para o sistema de produção em questão, os despachos foram considerados imediatos, isto é, entram na fabricação tão logo quanto seja possível. Contudo, preferiu-se citá-las para que sejam utilizadas em trabalhos futuros, onde esta característica seja necessária.

O Modelo Proposto Para Geração de Scheduling

4.4. O Scheduling

Segundo Arosio e Sianesi [1993], programação ou *scheduling* podem ser definidos como detalhados programas de operações a serem executados pelos recursos disponíveis no chão de fábrica. O *scheduling* é feito com base nas ordens de produção geradas, que devem ser inicialmente levadas para a programação, a fim de verificar a viabilidade de execução das mesmas. Chama-se a atenção para o fato de se tratar apenas da verificação do atendimento, pois a ordem já fora emitida. [RUSSOMANO, 1986]. Após esta fase de programação, as ordens são transmitidas para o chão de fábrica, fazendo com que as informações sejam enviados ao postos de trabalho de forma a proporcionar um fluxo lógico e preciso [CHO e WYSK, 1993]. Contudo muitas são as dificuldades para que este processo ocorra de forma simples o suficiente para que se possa aproveitar o máximo possível os recursos disponíveis no chão de fábrica.

Como resultado destas dificuldades, várias pesquisas tem sido desenvolvidas na áreas de controle da produção, mas até hoje, aplicações práticas têm sido insuficientes. A razão principal para isto é a dificuldade de se modelar e resolver problemas do cotidiano. Muitas das soluções surgidas se basearam no uso de heurísticas, aplicando-se a problemas e a situações bastante particulares [LECOMTE e DEJAX, 1991]. Contudo, tais estudos tiveram como virtude indicar procedimentos ou regras aplicáveis ao chão de fábrica. Alguns destes procedimentos são aproveitados neste trabalho como forma alternativa de priorização, isto é, são colocadas visando avaliar sua adequabilidade ao sistema operando sob determinada condição.

O Modelo Proposto Para Geração de Scheduling

4.5. O Gerador de Scheduling Proposto

Problemas de *scheduling* são considerados como problemas combinatórios de otimização. Hopfield e Tank [1985] demonstraram que em alguns destes problemas, o uso de redes neurais é possível. Contudo, modelos que utilizam tal técnica geralmente tendem a encontrar uma solução ótima local, convergindo geralmente para pontos próximos do estado inicial, isto é, a solução converge para um determinado ponto nas imediações do primeiro ponto testado, sem que sejam examinados outros pontos do sistema. Decidiu-se, aqui, utilizar um método otimizador que tenha como qualidade a capacidade de ser exploratório, isto é, que faça a pesquisa por pontos ótimos em vários pontos do espectro. Os Algoritmos Genéticos (GAs) foram então selecionados como método de otimização mais indicado para este tipo de problema [BEASLEY, 1993].

Na Figura 4.3 é mostrado o esquema de funcionamento do modelo proposto. Observa-se que além do uso de GAs, faz parte do sistema um módulo de simulação, que engloba tanto o modelo do sistema produtivo, propriamente dito, quanto os controladores e processadores, colocados para gerenciar o processo. O funcionamento do gerador pode ser descrito por partes, sendo a primeira relacionada aos ajustes dos parâmetros dos GAs. Após esta fase são criados os elementos da primeira geração de alternativas, testando em seguida estas mesmas alternativas no simulador. O simulador deve ser primeiramente carregado com dados externos relativos aos produtos e processos. Indivíduos são então selecionados para cruzamento, criando assim a geração seguinte de alternativas. Havendo a convergência, simula-se novamente com base no melhor ajuste conseguido, emitindo o relatório com as ordens de produção, serviço e montagem.

Deve-se salientar que os controladores utilizados são responsáveis em última análise pela execução da ordem emitida pelos processadores, isto é, são estes que realocam as peças nas

O Modelo Proposto Para Geração de Scheduling

filas, observando as prioridades colocadas pelos processadores centrais. Os controladores são ainda responsáveis pelo envio das informações relativas às máquinas por eles atendidas. Estas informações se referem ao número de peças na fila de entrada, na fila de espera por transporte e à carga total esperando processo. Nota-se que existem dois tipos de controladores, sendo um responsável pelo gerenciamento da fila de entrada da máquina, chamado aqui de controlador de fila, e outro responsável pelo gerenciamento das ordens de despacho das peças para transporte, denominado controlador de transporte.

Os processadores, conforme o que já foi dito, são os responsáveis pelo cálculo das prioridades das peças no aguardo das ordens de despacho. Portanto, os processadores funcionam como computadores que calculam conforme o estado do sistema a ordem das peças nas filas. Esta determinação é feita através do cálculo da pertinência das características das peças (tempo de entrada da peça no sistema, data de entrega, carga na fila de destino, tempo de processamento restante, etc.) e das regras disponíveis em relação à função objetivo. Da mesma forma que os controladores, existem dois tipos de processadores: um processador de filas e um processador de transporte, aos quais todos os respectivos controladores estão ligados. O esquema de ligação é mostrado na figura 4.4.

Os parâmetros difusos, usados pelos processadores, são calibrados, isto é, são ajustados visando máximo desempenho do sistema, pelos GAs. Os GAs fazem ajustes através da evolução das alternativas inicialmente determinadas. Isto é feito através do cálculo da adequação de cada alternativas à função objetivo, e da determinação de novas alternativas. A escolha pelo emprego de sistemas difusos deveu-se a potencialidade de aplicação do mesmo e pela facilidade de ajuste dos parâmetros difusos pelos GAs.

O Modelo Proposto Para Geração de Scheduling

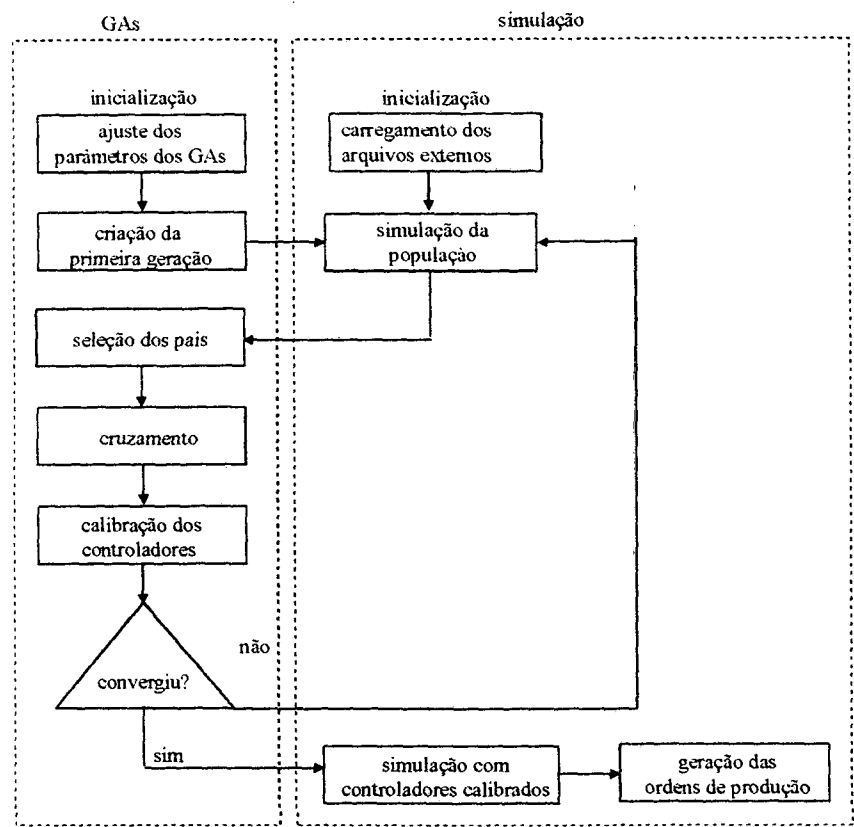


Figura 4.3. Esquema de funcionamento do gerador de scheduling.

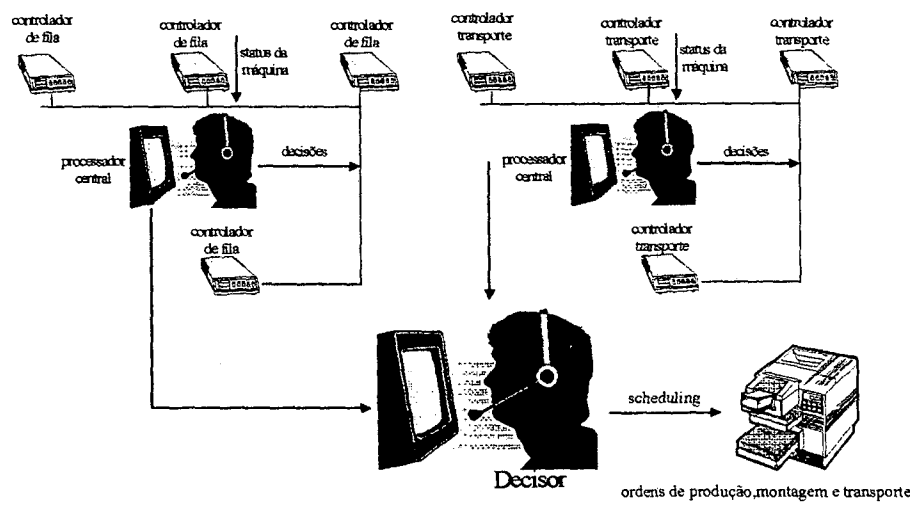


Figura 4.4. Esquema de ligação dos sistemas que compõe o modelo proposto. São vistos os controladores de processo e transporte, bem como os processadores centrais a eles conectados. O decisor é mostrado, sendo este quem irá tomar a decisão em última análise.

O Modelo Proposto Para Geração de Scheduling

4.6. A Simulação

A simulação desempenha papel fundamental no sistema proposto, interconectando as diversas ferramentas. Ainda, a implementação dos controladores durante a simulação permitiu que a tomada de decisão em termos operacionais fosse possível. Aqui a simulação possibilita, também, que implementações a nível de automação do processo sejam acrescentadas ao modelo. Chama-se atenção que sistemas de automação podem ser testados ou mesmo substituídos, sem haver perda das características dos mesmos, isto é, estes sistemas também podem ser simulados no modelo sem que existam na realidade. No caso específico da geração de *scheduling*, sistemas de automação que monitorem e se encarreguem da determinação das ordens, poderiam ser totalmente eliminados, uma vez que o sistema proposto seria o único responsável pela geração das ordens de serviço, de fabricação ou montagem.

Além da representação do funcionamento da planta, envolvendo a parte física e lógica do processo, incorporou-se também, ao modelo de simulação, regras provenientes de heurísticas. Estas regras foram implementadas diretamente nos controladores, sendo que os parâmetros relacionados ao controle difuso destas foram estabelecidos em arquivos externos, a fim de possibilitar o ajuste pelo algoritmo otimizador.

A seguir são descritos os principais componentes do modelo de simulação desenvolvido.

4.6.1. Projeto do Controlador

Neste item, serão discutidas as estratégias de controle operacional, bem como o projeto dos controladores utilizados

Primeiramente deve-se citar que decisões a respeito do planejamento devem ser tomadas em conjunção com políticas de

O Modelo Proposto Para Geração de Scheduling

controle operacional. O uso dos controladores permite que estratégias *on-line* sejam testadas em função do estado do sistema. Tal recurso possui vantagem óbvia, uma vez que permite tratamento diferenciado para cada tipo de situação. Isto proporciona um melhor desempenho do sistema conforme o objetivo pretendido.

4.6.1.1.

Controlador de Fila

O controlador de fila lida diretamente com seqüenciamento das peças relacionado ao problema de determinação da prioridade de fabricação das peças, isto é, do estabelecimento de qual peça será processada. Isto significa que o controlador será responsável pela ordenação da fila de entrada da máquina. Verificando o estado do sistema, o controlador liberará aquela peça que possibilite um melhor desempenho do sistema. Isto é feito pela seleção da peça que possuir maior pertinência em função de qualquer das regras, isto é, cada peça na fila terá sua pertinência calculada com base nas regras disponíveis. Note-se portanto que neste trabalho utilizar-se-á apenas um critério, ao mesmo tempo que a seleção se fará pela tomada do valor supremo das pertinências relacionadas a cada uma das regras (alternativas).

A representação esquemática do controlador de fila pode ser vista na figura 4.5. Resumidamente, pode-se observar na figura que conforme o cálculo das pertinências vindo do processador de fila, o controlador selecionará a peça com maior pertinência e transmitirá a informação ao respectivo módulo interno (cada módulo é relacionado a uma das regras) que será responsável pela execução da ordem estabelecida. Parâmetros relacionados ao estado da máquina são transmitidos constantemente ao processador de fila que levará os mesmos em consideração durante a tomada de decisão, através da saída indicada pelo rótulo *status*.

O Modelo Proposto Para Geração de Scheduling

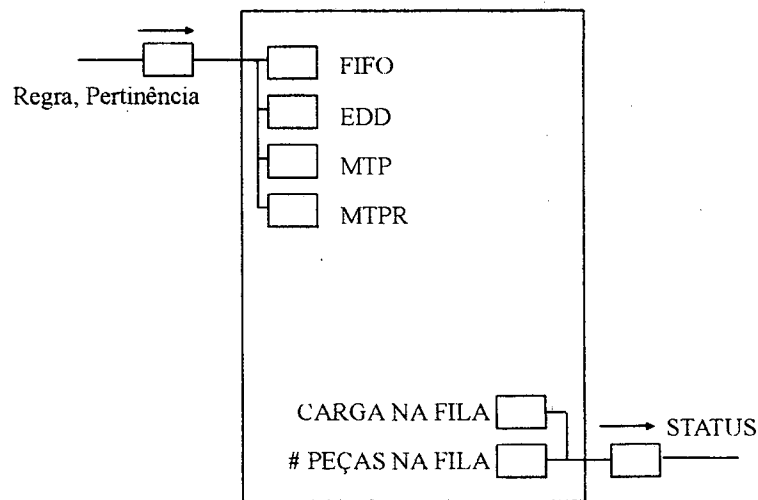


Figura 4.5. Representação esquemática do controlador de fila.

Em termos lógicos, o funcionamento do controlador e processador de fila pode ser representado conforme figura 4.6. Nota-se que o controlador é responsável, sempre que houver liberação de peça para processo, por mandar as informações de cada peça ao processador de fila. Isto é feito através da retirada de todas as peças da fila, uma a uma. A cada retirada, o processador calculará a pertinência de cada peça com relação às regras de seqüenciamento utilizadas, devolvendo-as novamente para a fila. Ao fim deste processo o controlador escolherá a peça que possuir maior pertinência e executará a retirada da mesma da fila, a fim de liberá-la para processo.

Em geral, as regras de seqüenciamento são relacionadas as características das peças em processos, referindo-se aos tempos de processo, data de entrega, número de operações ou tempo de chegada da peça. Contudo, existem regras que não são baseadas apenas nos atributos das peças. Regras que levam em consideração o estado das filas no sistema são exemplos deste tipo de regras. A seguir serão discutidas quatro regras utilizadas neste trabalho.

O Modelo Proposto Para Geração de Scheduling

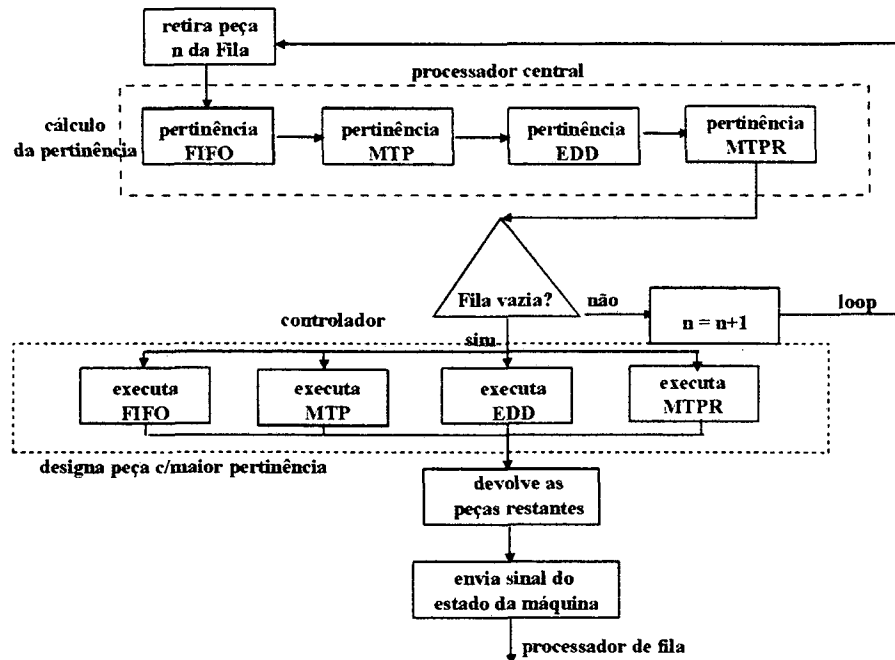


Figura 4.6. Lógica de funcionamento do controlador de fila e do processador central

4.6.1.1.1. Regras para a Priorização de Peças no Buffer das Máquinas

Estas regras tratam da priorização das peças que se encontram nos *buffers* das máquinas aguardando para serem processadas. A prioridade é designada a uma peça, quando mais de uma se encontra aguardando. As regras escolhidas são:

- *Menor tempo de processo* (MTP): tem sido uma das regras de prioridade mais estudadas para *job-shop*. Esta regra seleciona a peça que possa ser completada no menor tempo possível. Em sistemas de uma máquina, foi demonstrado que esta é a regra de seqüenciamento ótima para maximização do *throughput* [CONWAY E MAXWELL, 1962]. Contudo, em sistemas mais complexos foram encontrados alguns exemplos onde sua aplicação não era tão eficiente [STECKE E SOLBERG, 1981; SHANKER E

O Modelo Proposto Para Geração de Scheduling

TZEN, 1985]. Em geral esta regra pode ser considerada eficiente.

- *Menor Tempo de Processamento Restante (MTPR)*: MTPR seleciona para processamento, a peça cujo tempo de processamento restante seja o menor.
- *First in First out (FIFO)*: A prioridade de fabricação é determinada pela ordem de chegada da peça na máquina. O desempenho desta regra é considerado deficiente em comparação com outras regras em aplicações em job-shop [BLACKSTONE e outros., 1982]. Contudo alguns estudos indicaram que o desempenho da FIFO não é significativamente diferente de outras [STECKE e SOLBERG, 1982].
- *Earliest Due Data (EDD)*: Determina-se a prioridade com base no tempo previsto para entrega, a fim de se minimizar o tempo em atraso do produto final.

4.6.1.2.

Processadores de Fila

Os processadores são responsáveis pelo cálculo das pertinências usadas na seleção das peças feita pelos controladores. Basicamente, o processador calcula a partir do estado do sistema e através de informações vinda dos controladores, qual seria a pertinência daquela regra relacionada à peça naquele momento. A figura 4.7 esquematiza o processador. A decisão está baseada em critérios difusos.

O Modelo Proposto Para Geração de Scheduling

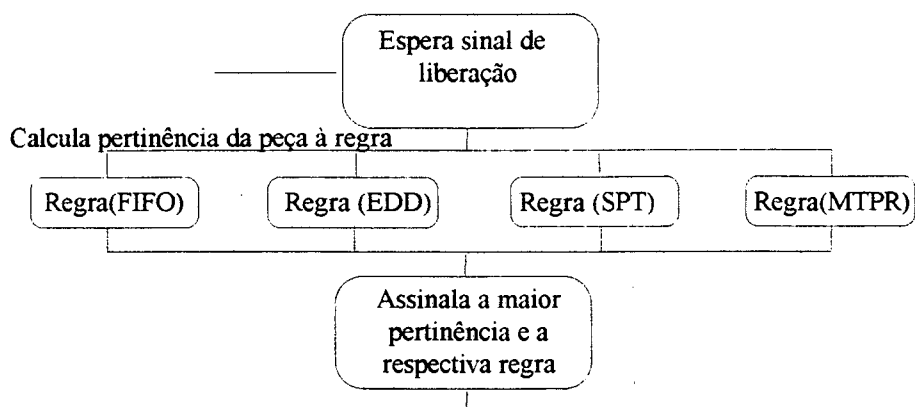


Figura 4.7. Cálculo das pertinências. O processador com base nas informações fornecidas pelo sistema calcula a pertinência de cada peça com relação a cada regra. Por fim assinala, através de atributos, a regra selecionada e sua respectiva pertinência.

4.6.1.3.

Controladores e Processadores de Transporte

Os controladores e os processadores de transporte estão diretamente ligados à política de despacho de veículos. Funcionam de maneira semelhante ao controlador de fila e ao processador de fila, diferindo apenas nos algoritmos de implementação das regras utilizadas.

A representação esquemática do controlador de transporte pode ser vista na figura 4.8. Nota-se que os parâmetros difusos são repassados diretamente ao controlador pelo GAs. Ainda, o *status* da máquina é transmitido constantemente ao processador de transporte, bem como é alocado um espaço na fila do transportador sempre que alguma peça seja colocada a disposição para ser transportada.

O funcionamento lógico do processador de transporte pode ser descrito pela figura 4.9. Observa-se que o processador testa cada uma das peças que estão disponíveis na fila aguardando transporte, calculando para cada uma o valor de pertinência, independente da regra. O cálculo é feito pelo processador central, que dispõe de todas as informações necessárias, inclusive aquela

O Modelo Proposto Para Geração de Scheduling

informações gerais sobre o sistema. O processador escolhe então aquela que tiver o maior valor de pertinência e executa a liberação da peça para transporte.

As regras usadas são examinadas a seguir.

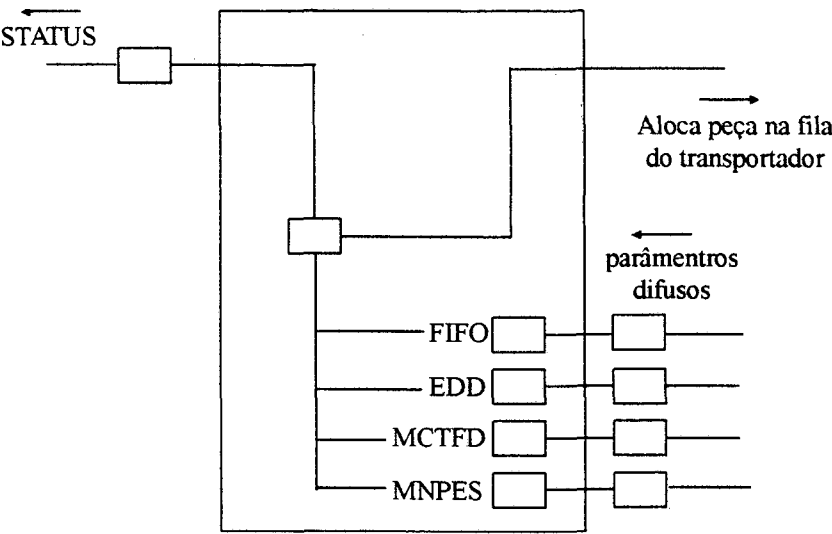


Figura 4.8. Representação esquemática do controlador de transporte..

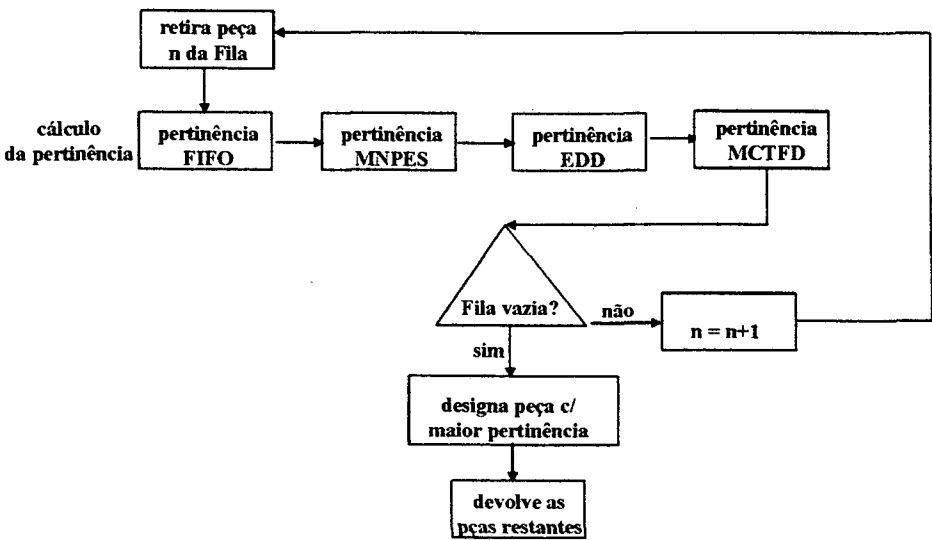


Figura 4.9. Lógica de funcionamento do processador de transporte.

O Modelo Proposto Para Geração de Scheduling

4.6.1.3.1. Políticas para o Despacho de Veículos Transportadores

As regras de despacho de veículos transportadores são necessárias quando uma peça precisa ser transportada de uma para outra máquina. A prioridade deve ser designada quando mais de uma peça encontra-se aguardando por transporte no momento em que um veículo torna-se disponível. Estas regras podem ser dependentes das características das peças ou do estado do sistema [BOZER, 1989;. LEUNG, 1990, GUPTA, 1992; MAHESHWARI, 1992; SABUNCUOGLU, 1992; YIM, 1993]. A seguir são descritas as regras utilizadas:

- *Menor Carga de Trabalho na Fila de Destino (MCTFD)*: A prioridade é dada à peça cuja fila de destino encontra-se com a menor carga de trabalho. A carga é medida pela soma dos tempos de processamento das peças que se encontram naquela fila. O objetivo desta regra é reduzir o desbalanceamento na carga das máquinas.
- *Earliest Due Data (EDD)*: Determina-se a prioridade com base no tempo previsto para entrega, a fim de se minimizar o tempo em atraso do produto final.
- *Máximo Número de Peças Esperando para Sair (MNPES)*: Esta regra determina que a prioridade deva ser dada a peça que se encontrar numa máquina, cuja fila de peças esperando para serem transportadas a outra estação, seja a maior.
- *FIFO: (First in First out)*: A prioridade é determinada de acordo com a ordem de solicitação de transporte das peças.

4.6.2. Parâmetros Utilizados no Modelo

Determinação do Produto: uma vez gerada a ordem de serviço para o produto, o simulador, através de um Banco de Dados, explode esta ordem em ordens de fabricação das peças componentes.

O Modelo Proposto Para Geração de Scheduling

Número de Operações: cada peça necessita um determinado número de operações para completar seu processamento. No modelo preferiu-se codificar as peças, a fim de facilitar a inserção de novas peças no modelo de simulação. Os dados se encontram armazenados em um arquivo externo, gerado por um banco de dados.

Tempo de Processamento: cada peça necessita de uma determinada quantidade de tempo em cada operação. Estes dados também estão vinculados ao código da peça, sendo os tempos armazenados em arquivos externos.

Número e Velocidade dos Transportes: a decisão sobre o número de veículos transportadores a serem utilizados em um sistema, objetivando assegurar uma particular taxa de produção, depende, dentre outras variáveis, do arranjo físico do sistema e de sua programação. Esta decisão é típica da fase de detalhamento do projeto, na qual um número razoável de alternativas de programação da produção devem ser testadas (simuladas) em relação a um determinado arranjo físico, para que se possa concluir algo, a partir de inferências sobre os resultados do modelo testado. Quanto as velocidades, estas podem ser facilmente alteradas pelo analista durante os procedimentos de avaliação do desempenho dos sistemas.

Tamanho dos Buffers: Considerou-se a capacidade dos *buffers* como infinita. Isto foi feito visando determinar tanto os gargalos quanto o tamanho ideal dos *buffers*. Contudo a determinação de capacidades finitas para os mesmos pode ser feita sem maiores dificuldades, bastando para isso ajustar os valores no modelo de simulação.

O Modelo Proposto Para Geração de Scheduling

4.6.3. Modelo de Simulação Proposto

Os parâmetros básicos de entrada do modelo foram definidos no item anterior. Para que o sistema possa funcionar é preciso, no entanto, que para cada tipo de peça a ser processada exista uma seqüência de operações previamente determinada, a qual está, normalmente, atrelada a uma seqüência de visitas de máquinas, de tal forma que no momento de sua entrada no sistema, seja possível determinar seu destino. No modelo proposto, a determinação da seqüência de visitas das máquinas é estabelecida *a priori*, uma vez que não são consideradas máquinas alternativas.

A simulação é iniciada com o sistema vazio e todos os equipamentos disponíveis. Subseqüentemente, as peças entram no sistema à medida em que os pedidos são feitos. As ordens de fabricação das peças são liberadas de acordo com a composição do produto. Após completar todo seu processamento, as peças são levadas para montagem, ou caso sejam componentes de outras peças, aguardam na máquina até que o restante dos componentes cheguem para dar continuidade ao processo.

O tempo de execução de uma simulação depende, usualmente, dos objetivos do analista. Na grande maioria dos processamentos ligados a projetos de manufatura, este tempo está associado ao ciclo de produção de um determinado conjunto de peças. Outras vezes, este tempo relaciona-se com os turnos de produção do próprio sistema, como, por exemplo, quando se deseja saber qual seria a produção alcançada, no período de uma semana, com turnos diários de 8 horas. No caso em estudo, simulou-se por um tempo determinado, sem ter havido preocupação com a execução das ordens de produção.

O Modelo Proposto Para Geração de Scheduling

4.6.3.1. Hipóteses Associadas ao Modelo

Os modelos de simulação, em geral, não requerem tantas hipóteses quanto os modelos analíticos. No entanto, o desenvolvimento de qualquer tipo de modelo, necessita um claro delineamento de suas fronteiras. Além disso, pode não ser apropriado, nem mesmo viável, incorporar todo o tipo de detalhamento ao modelo de simulação. Assim, para manter o esforço de modelagem sob controle é necessário fazer algumas hipóteses razoáveis sobre o sistema. No modelo desenvolvido, as seguintes hipóteses foram feitas:

- O tamanho de *buffer* para todas as máquinas é infinito;
- Todas as matérias primas das peças a serem produzidas estão disponíveis, isto é, não há atraso devido a falta de matéria prima;
- Os veículos de transporte permanecem na estação em que foram liberados, até que haja outra solicitação.

4.6.3.2. Controle Lógico das Máquinas

O controle lógico das máquinas, está totalmente associado às peças que podem estar sendo processadas ou aguardando para serem processadas ou transportadas. Se mais de uma peça aguarda por processamento, então a regra de prioridade de seqüenciamento é aplicada para selecionar uma peça. A regra é selecionada pelo processador de filas, sendo repassadas para os controladores de fila, que serão os responsáveis pela determinação da peça a ser levada para a máquina. A figura 4.11 ilustra esta lógica.

O Modelo Proposto Para Geração de Scheduling

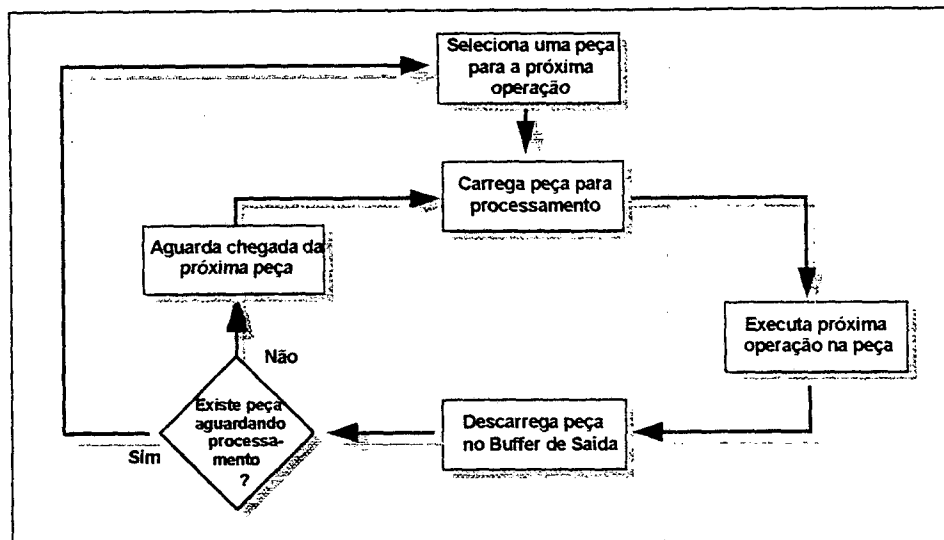


Figura 4.10. Controle Lógico das Máquinas

4.6.3.3. Medidas de Desempenho

As medidas de desempenho operacional para a avaliação simulada de sistemas de produção podem ser as mais diversas. As principais medidas de desempenho operacional utilizadas neste estudo são:

- o tempo de passagem;
- o tempo do ciclo produtivo;
- a taxa de utilização das máquinas;
- a taxa de utilização dos transportadores;
- o tamanho das filas nas máquinas

O tempo de passagem é definido como o tempo que uma peça permanece no sistema, iniciando-se sua contagem no momento em que esta é liberada para o sistema, isto é, na sua entrada na primeira máquina do seu roteiro até sua saída para montagem ou para casamento com outra(s) peça(s).

O tempo do ciclo de produção, é definido pelo tempo total para a manufatura de todas as peças requeridas. Nos ambientes produtivos sob investigação, somente um número finito de peças

O Modelo Proposto Para Geração de Scheduling

é produzido em cada ciclo. Da literatura [MAHESHWARI, 1992], fica comprovado que medir o tempo do ciclo produtivo, é análogo a medir-se a utilização do sistema como um todo ou sua taxa de produção.

A taxa de utilização das máquinas é determinada pela relação entre o tempo que cada máquina passa efetivamente processando peças e o tempo total do ciclo produtivo.

A taxa de utilização dos transportadores é determinada pela relação entre o tempo que cada transportador passa movimentando-se, carregado ou vazio e o tempo total do ciclo produtivo. Outra taxa que pode ser utilizada é a de utilização efetiva dos transportadores, na qual somente o tempo de uso com carga é computado.

O tamanho das filas das máquinas é fornecido como resultado de uma média ponderada pelo tempo, do número de elementos que se encontrem, ao longo da simulação, naquelas filas. É uma importante medida, diretamente relacionada com detecção de gargalos.

4.7. O Otimizador O método de otimização escolhido foi baseado no uso de GAs. Faz-se necessário explicitar que os GAs são otimizadores matemáticos, sendo portanto responsáveis apenas pela maximização da função objetivo estabelecida. No caso do modelo proposto, esta função é dada pela resposta do simulador em função das alternativas colocadas.

Os algoritmos otimizadores utilizados foram desenvolvidos em linguagem C++, utilizando o compilador Borland C / C++ 3.0. Este programa é conectado ao Arena, através de comandos de acesso ao sistema operacional MS-DOS 6.2 da Microsoft e por

O Modelo Proposto Para Geração de Scheduling

arquivos externos usados tanto pelo Arena quanto pelo programa de otimização. A listagem encontra-se no ANEXO II.

O esquema de funcionamento dos GAs é mostrado na figura 4.12. Chama-se a atenção para a inicialização do sistema, que necessita da definição de alguns parâmetros. Estes parâmetros são relacionados às características da população (tamanho da população, tamanho das *strings*, probabilidade de ocorrência de mutação e *crossover*), à geração de número aleatórios (semente), e ao tamanho do experimento (número de gerações).

Uma vez definidos os parâmetros iniciais, os GAs se encarregam de gerar alternativas para análise. As alternativas iniciais (primeira geração) são baseadas exclusivamente na geração aleatória de indivíduos. A partir destes indivíduos, o sistema procurará otimizar o desempenho baseado no mecanismo de seleção natural, isto é, indivíduos serão gerados a partir de sorteios, que escolherão os pais. Observa-se que indivíduos com maior *fitness* terão maior probabilidade de cruzamento. Isto faz com que alternativas que não obtiverem sucesso, naturalmente desapareçam, enquanto alternativas, que apresentem um bom desempenho, evoluam, gerando novas alternativas, mantendo parte significativa das características dos pais.

Explicado o mecanismo de reprodução, torna-se necessário examinar o mecanismo relacionado ao *crossover* e à mutação. O mecanismo de *crossover* aqui utilizado foi o chamado *crossover* 2. Este mecanismo é baseado na quebra dupla do cromossomo, isto é, o cromossomo é seccionado em duas partes, havendo troca de material genético entre os indivíduos nestas seções. Este algoritmo foi utilizado por ter melhor desempenho, isto é, convergir mais rapidamente para o valor ótimo. Quanto ao mecanismo de mutação, este é baseado na probabilidade de ocorrência do mesmo. Havendo a mutação, a rotina se encarrega

O Modelo Proposto Para Geração de Scheduling

de escolher um dos alelos, alterando sua paridade. A figura 4.14 esquematiza a geração de indivíduos.

A nova população gerada conforme as regras acima, ainda colocada no vetor *newpop* é transferida então para o vetor *oldpop*, desencadeando assim o processo geração de nova população. Este processo acaba, quando os indivíduos de uma população apresentarem convergência, isto é, 95% dos indivíduos da população apresentarem mesmo material genético (cromossomos semelhantes). Neste ponto, o valor ótimo dos parâmetros dos controladores é obtido.

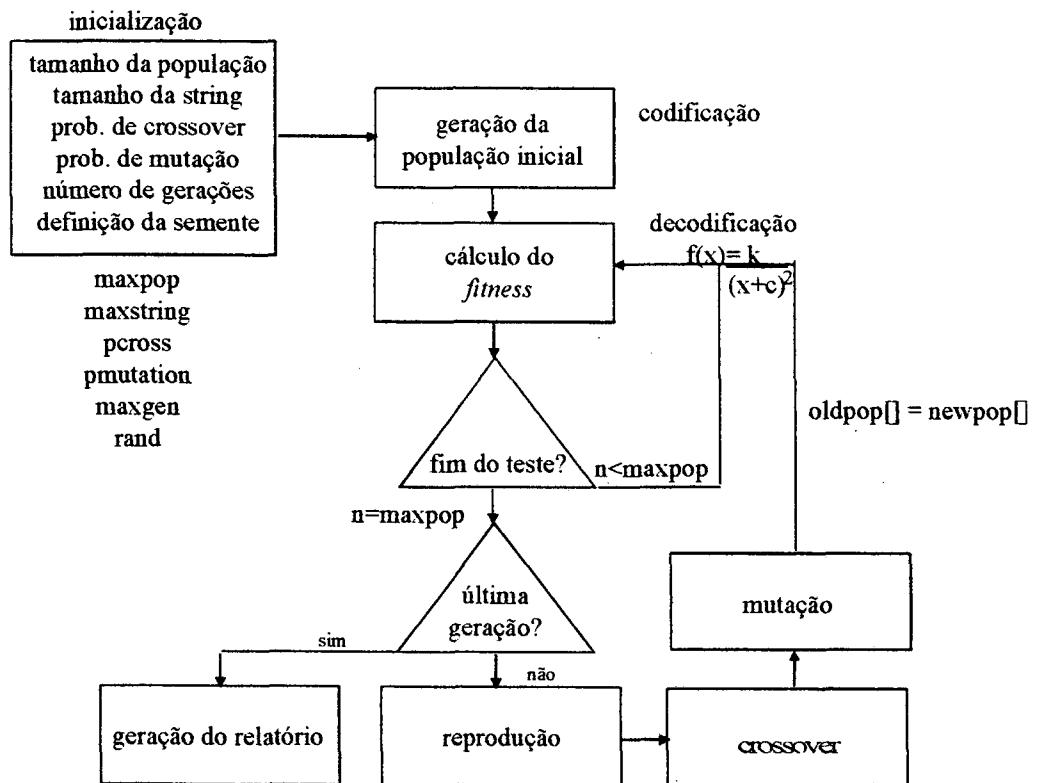


Figura 4.11. Esquema de funcionamento dos GAs.

O Modelo Proposto Para Geração de Scheduling

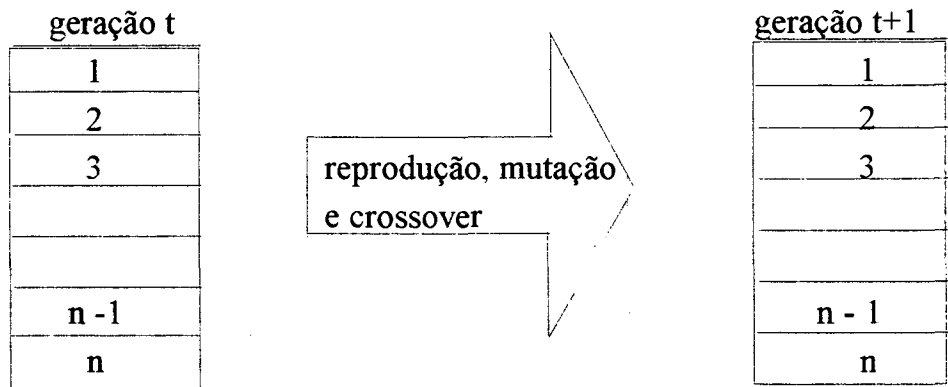


Figura 4.12. Representação esquemática dos operadores genéticos usados nos GAs e da população antiga e da nova ("non overlapping").

Com relação a função objetivo, esta é calculado através do parâmetro repassado pelo simulador aos GAs. Na aplicação desenvolvida trabalhou-se apenas com a minimização dos valores do tempo em atraso, relativos aos tempos de entrega das peças, em termos médios e máximos. Salienta-se que poderia ser usado qualquer outro critério, como por exemplo, maximização de lucro, minimização de custo, minimização do tempo total de processo, etc. A função objetivo é vista abaixo:

$$\text{função objetivo} = k / (f(t))^2 \quad (\text{eq. 4.1})$$

onde,

- $f(t)$ é o valor repassado aos GAs pelo Arena (atraso médio ou máximo);
- k é uma constante.

Utilizou-se o inverso da função por se tratar explicitamente de um método de minimização. Elevou-se ao quadrado a função, visando obter valores sempre positivos para a função, e para tornar possível a procura pelo valor alvo, isto é, possibilitar que o sistema consiga um valor zero para o atraso. A constante k é

O Modelo Proposto Para Geração de Scheduling

usada apenas para tornar a grandeza dos valores da função objetivo mais facilmente representáveis, não tendo assim nenhuma influência sobre o resultado.

4.8. Sumário

Neste capítulo apresentou-se os principais elementos envolvidos no problema do planejamento e controle operacional dos sistemas de manufatura, bem como o modelo de geração de *scheduling* proposto. O esquema de ligação e as lógicas externas dos módulos envolvidos (modelo de simulação, controladores, processadores e GAs) foram detalhadamente descritos. As políticas operacionais, implementadas através de heurísticas foram também discutidas neste capítulo, bem como as hipóteses e limitações consideradas. Por último foi mostrado o algoritmo otimizador utilizado, no caso os GAs, bem como as particularidades de seu funcionamento.

APLICAÇÃO DO MODELO PROPOSTO

Neste capítulo será descrita uma aplicação do modelo proposto, bem como os resultados obtidos, a fim de comprovar a utilidade do mesmo. Procurar-se-á também descrever os passos seguidos para o desenvolvimento do projeto, buscando auxiliar na implementação de novos modelos e sistemas.

5.1. Empresa Escolhida

A empresa em questão é fabricante de produtos não seriados de grande porte, isto é, os produtos fabricados são produzidos em baixo volume com tempos de ciclo elevados, normalmente dados em horas. O sistema de produção é convencional, sendo a produção organizada por estações de trabalho. O departamento de PCP é encarregado de gerir a produção, fazendo o planejamento das operações através da observação da disponibilidade de máquinas e do acompanhamento de atividades, que é feito através de gráficos de Gantt. Este procedimento serve para indicar tempos de entrega de peças e do produto final, verificando-se que apenas as atividades críticas são planejadas. As demais atividades vão entrando na produção conforme sejam os tempos de entrega e as folgas das máquinas.

A demanda é instável, mas todos os pedidos feitos são considerados firmes, isto é, uma vez expedida a ordem não haverá cancelamento da mesma, nem se produzirá para formação de estoque.

Aplicação do Modelo Proposto

Existem 24 estações de trabalho, compreendendo estações de fabricação e montagem. A mão de obra é associada a cada estação de trabalho, não sendo possível a atuação de operários em mais de uma estação. Ao redor de 1000 peças são fabricadas para um a carteira de 100 tipos de produtos. A demanda se situa em torno de 200 produtos/ano.

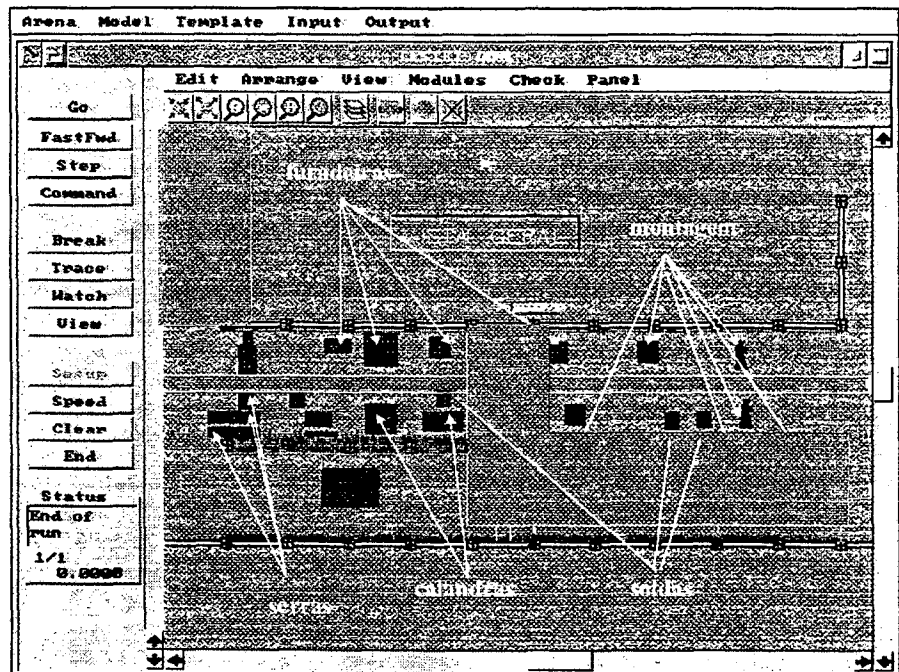


Figura 5.1. Layout da fábrica. São mostradas as máquinas e as estações de trabalho. O fluxo segue da esquerda (onde se encontra a entrada de matéria prima) para a direita onde está a montagem. No vão superior o transporte é feito por uma ponte, auxiliada por vagonetes. No vão inferior só há transporte por vagonetes.

Aplicação do Modelo Proposto

5.2. Modelagem do Sistema Produtivo

O desenvolvimento do modelo foi dividido em etapas, a fim de que o processo pudesse ser conduzido com a maior rapidez possível, ao mesmo tempo que assegurasse a consecução dos objetivos pretendidos. Apesar de colocadas em seqüência, estas etapas não foram desenvolvidas nesta ordem cronológica, isto é, algumas das etapas puderam ter seu início adiantado, sendo feitas em paralelo com outras atividades, ao mesmo tempo que algumas atividades tiveram seus cronogramas atrasados.

Estas etapas foram assim divididas:

- coleta e análise de dados;
- análise de fluxos;
- modelagem do sistema;
- análise de alternativas;

5.2.1. Coleta e Análise de Dados

Dados determinísticos foram utilizados, em virtude da dificuldade de amostrar os tempos de processo. Tal dificuldade foi gerada pelas características específicas da empresa, que como fabricante de produtos não-seriados, não tornou possível a existência de quantidades razoáveis de dados para uma estimativa em termos probabilísticos. Contudo, confiou-se satisfatoriamente nos dados passados, e afirma-se com segurança que estes dados representam razoavelmente a realidade. Encontram-se os mesmos no Anexo IV.

5.2.2. Análise de Fluxos

Os fluxos dos produtos, ao redor de 1000 (30 famílias de peças), foram feitos a partir dos mapas de produção da empresa. O agrupamento em torno de famílias de peças permitiu a utilização de um único roteiro para produtos da mesma família. No Anexo IV são mostrados os roteiros.

Aplicação do Modelo Proposto

A maioria dos fluxos seguem uma sequência indo da esquerda para a direita (vide figura 5.1) e envolvem geralmente o corte de chapas (feito nas guilhotinas ou máquinas de oxicorte), a serração (feita nas serras), a furação (executada nas furadeiras radiais), a conformação de chapas (feita nas calandras), a soldagem de peças (feito por soldagem manual ou por soldagem através de máquinas TIG) e montagem de componentes (feitos em separado, conforme o tipo de peça nas estações de trabalho). Finda a etapa de fabricação dos grandes grupos de componentes (peças grandes, como por exemplo corpo de caldeira, rotor de gerador, vasos de armazenamento, etc.), estas partes são transportadas para a montagem onde são unidas para constituírem o produto final. No Anexo IV são mostrados os fluxos das peças e no Anexo V os componentes de cada produto.

5.3. Modelagem do Sistema

O modelo foi desenvolvido a partir de dados e lógicas existentes na unidade, fornecidos pela empresa. Esta modelagem procurou tornar possível, através do aumento da flexibilidade do modelo, o estudo de várias alternativas, possibilitando o atendimento ao objetivo proposto, isto é, desenvolver um modelo para geração de *scheduling* que possa ser rapidamente alterado conforme ocorram mudanças na planta.

5.3.1. Características do Modelo

As seguintes características foram consideradas relevantes para o modelagem do sistema:

a) máquinas: tempos de execução da tarefa foram considerados como tempo de processamento mais o tempo de ajuste das ferramentas (tempo de *setup*) Visando garantir ao mesmo tempo a aderência do modelo a realidade e a economia em

Aplicação do Modelo Proposto

tempo no seu desenvolvimento, uma série de características foram ignoradas em detrimento àquelas mais relevantes. Equipamentos de menor porte, como lixadeiras, vagonetes, soldas, etc., tiveram seus tempos de processo englobados a outros. Também não foi considerada a hipótese de existência de máquinas alternativas. No que diz respeito à lógica, chama-se a atenção para o fato de que peças, que devam ser montadas em conjunto com outras peças componentes, ficam aguardando até que todos os componentes necessários estejam disponíveis na máquina para então dar continuidade ao processo. Por fim, não foram consideradas limitações de espaço para colocação de peças, tanto na saída, quanto na entrada. Isto foi feito com o intuito de verificar a existência de desbalanceamentos e de gargalos no sistema;

b) mão de obra: nas estações de trabalho relacionadas a fabricação, os tempos de processo foram dados em horas. Nas estações de trabalho relacionadas às etapas de montagem os tempos de processo foi dado em homem/hora. Isto significa que o aumento de mão de obra nestas últimas influencia o tempo de processo, reduzindo diretamente o tempo gasto para execução. Condiçãoou-se ainda a liberação da estação para novo processo à finalização do produto em fabricação, isto é, toda a mão de obra disponível na estação é utilizada em apenas um produto, não sendo possível o processo de mais de uma peça por estação.

c) transportes: não foram considerados tempos de preparação e liberação da carga, bem como o tempo de posicionamento da talha. Observa-se ainda que quando da não utilização de equipamentos de transporte, isto é, quando a peça é transportada manualmente, o tempo de percurso é dado por 3 min. A velocidade da ponte foi estimada em 7m/min., não sendo considerandas a aceleração e a desaceleração. Como última observação, chama-se a atenção para o fato de que não

Aplicação do Modelo Proposto

foi simulada a operação dos vagonetes, pois os mesmos equipamentos foram considerados auxiliares ao processo e normalmente estão disponíveis, tendo uma influência bastante pequena no desempenho do sistema;

d) demanda: foi calculada com base nos pedidos efetuados em um período determinado (5000 minutos de produção). Este intervalo de simulação conhecido como “janela de simulação” serve para indicar o período considerado na definição do *scheduling*, isto é, foi feita a simulação visando a programação das máquinas para os próximos 5000 minutos. As estatísticas foram iniciadas após um período de 500 minutos, a fim de aquecer o sistema (“*warm-up*”), antes de começar a coletar as estatísticas.

e) turnos de trabalho: foi considerada a operação das máquinas apenas em um turno. Este turno começa às 6:00 h. e vai até as 10:00 h., quando pára para almoço. Reinicia às 10:15 h. e termina às 15:00 h, totalizando assim um turno de 8:45 h.

5.4. O Desenvolvimento do Modelo de Simulação no ARENA

O modelo de simulação feito em linguagem Siman V do pacote de simulação ARENA 1.2 (Systems Modeling Corp.) foi dividido nas seguintes etapas:

a) determinação dos parâmetros de inicialização do sistema: nesta etapa foram determinadas as características iniciais do sistema, como por exemplo, a determinação das capacidades de máquinas, entrada de pedidos, tempo de simulação, *warm-up*, posicionamento da ponte, bem como de suas características (velocidade, aceleração e desaceleração), códigos de peças, entre outros;

Aplicação do Modelo Proposto

b) desenvolvimento de rotinas de carregamento de dados via arquivos externos: visando uma maior flexibilidade, uma vez que os produtos estão sujeitos a modificações, assim como há normalmente introdução de novos produtos. As características relacionadas a tempos de processo, roteiros e componentes de produto foram então colocados em um arquivo externo gerados por um banco de dados (ACCESS da Microsoft) e estão listados nos Anexos IV e V. Estes arquivos foram convertidos em arquivos tipo ASCII para possibilitar que o simulador pudesse lê-los. Uma rotina no início da simulação é responsável pela leituras destes dados;

c) criação da rotina de explosão do pedido: uma vez gerado o pedido do produto, um rotina se encarrega de explodi-lo, isto é, o pedido é dividido em ordens de fabricação de peças, componentes do produto. Esta mesma rotina manda as ordens de fabricação para as máquinas encarregadas de executar as operações. Caso não haja necessidade de esperar por outro componente, esta peça passa imediatamente para a fila de espera para fabricação, caso contrário aguarda até que seu(s) componente(s) cheguem à máquina;

d) modelagem das máquinas: é a parte mais complexa da modelagem do sistema, devido as lógicas envolvidas. Foi necessário desenvolver rotinas que garantissem que o produto a ser fabricado somente fosse mandado para a fabricação após todos os componentes necessários chegarem. Rotinas relacionadas ao transporte foram criadas e embutidas em cada máquina.

e) modelagem dos controladores e processadores de fila: utilizou-se quatro regras nos controladores (MTP, EDD, FIFO, MTPR). Cada regra possui sua respectiva rotina para seleção da entidade a ser retirada da fila. O critério de seleção da melhor regra é dado pela expressão abaixo, sugerida por Gupta

Aplicação do Modelo Proposto

e outros [1993], que leva em consideração sobretudo a folga existente:

$$\begin{aligned} f(x) &= \text{função de avaliação} \\ &= (\text{Tempo de Entrega} - T_{\text{now}}) / (\text{Tempo de Processo} \\ &\quad \text{Restante} * k) \end{aligned} \quad (\text{eq. 5.1})$$

onde,

- k = constante para tempo de fluxo estimado, considerada igual a 1
- T_{now} = igual ao tempo atual de simulação
- x = alternativa = valor da *string*

Gupta [1993] utilizou esta função para calcular uma regra de despacho denominada por ele de CRATION. Aqui, esta regra é usada para determinar a melhor regra aplicável, servindo como parâmetro de avaliação da pertinência de cada regra, isto é, os valores das pertinências são funções do valor da eq. 5.1.

176
A “defuzzification” do resultado, também implementada na modelagem do controlador, é feita pelo processo de máxima pertinência, isto é, é escolhida a peça que possuir maior pertinência, aplicando-se o critério de tomada de decisão definido no item 3.3.4 (Tomada de Decisão Multi-Critério Difusa). Salienta-se que apenas um critério foi utilizado, sendo este dado pela eq. 3.3. Assim não foram considerados pesos para os critérios ($w=1$). A seleção se fará com base no valor absoluto do atributo pertinência, e não por regra, pois o máximo valor do atributo é que indicará a regra a ser aplicada. Chama-se a atenção para o fato de que o método de “defuzzification” pelo cálculo do centro de massa do trapézio, embora produza resultados melhores, não foi utilizado, devido a dificuldade de se executar o cálculo de integrais no ARENA.

Aplicação do Modelo Proposto

f) modelagem dos controladores e processadores de transporte: semelhante a modelagem dos controladores de fila, os controladores de transporte se diferenciam apenas pelas regras utilizadas: FIFO, EDD, MCTFD e MNPES. A função de avaliação foi mantida, dada pela eq. 5.1;

g) estatísticas: foram observadas estatísticas com relação a ocupação de máquinas e transporte, tamanho de filas, tempo de fluxo e atraso na fabricação.

O modelo desenvolvido no ARENA 1.2. em linguagem SIMAN V é listado no ANEXO I.

5.5. Experimento Procurou-se demonstrar através da aplicação, que o método é viável para problemas de otimização. No caso em estudo objetivou-se a minimização do tempo em atraso, isto é, a diferença entre o tempo previsto para a entrega e o tempo gasto efetivamente para cumprir a ordem de fabricação.

Uma vez desenvolvido o modelo, escolhida a função a ser otimizada, partiu-se para a definição dos parâmetros relativos aos GAs. Os valores são mostrados na tabela 5.1

parâmetro	variável	valor
tamanho da população	maxpop	18
tamanho da <i>string</i>	maxstring	24
prob. de <i>crossover</i>	pcross	.85
prob. de mutação	pmutation	.001
número de gerações	maxgen	20

Tabela 5.1. *Parâmetros iniciais utilizados.*

Aplicação do Modelo Proposto

Cabe colocar que estes valores utilizados na determinação dos parâmetros dos GAs são importantes para a efetiva busca do valor ótimo. Valores baixos para o tamanho da população (alternativas estabelecidas a cada geração) pode impedir que haja possibilidade de estabelecer um razoável número de alternativas, assim como um elevado número de indivíduos na população pode levar a redundâncias na pesquisa, e como consequência um elevado gasto de tempo.

O tamanho da *string* foi estabelecido pelo número de alternativas disponível para cada parâmetro (8 valores, compreendidos no intervalo de 0 a 7) (para listagem das alternativas vide tabela 5.2) e pelo número de regras utilizadas (4). Sabendo-se que para cada regra são necessários dois parâmetros (um valor máximo e outro valor mínimo), uma vez que estes valores são o bastante para descreverem as curvas de uma função triangular, com valor máximo no ponto médio, e que o valor 7 em binário necessita de três algarismos para representação (7 em binário é igual a 111), serão necessários, tem-se que o tamanho da *string* é dado por:

$$2 \text{ (parâmetros)} \times 4 \text{ (regras)} \times 3 \text{ (algarismos binária)} = 24$$

(eq. 5.2)

Um exemplo de *string* associado ao controlador de fila é mostrado abaixo,

MTP		EDD		MTPR		FIFO	
min	máx	mín	máx	mín	máx	mín	máx
000	111	000	111	000	111	000	111

Neste exemplos, o valor numérico em decimal passado pelos GAs ao simulador é $x = 1864135$ e significa que todos os parâmetros de todos as regras são ajustados para -10 (valor mínimo) e 10 (valor máximo).

Aplicação do Modelo Proposto

As probabilidades de ocorrência de *crossover* (*pcross*) e mutação (*pmutation*) podem influir significativamente na convergência dos dados. Geralmente os valores de *pcross* se situam entre 0,6 e 1,0 e os valores de mutação estão entre 0,0001 e 0,001.

Com os valores colocados, precisou-se de até 20 gerações para que os valores convergissem e o valor ótimo fosse dado como achado. Poderia-se tentar novas alternativas com valores diferentes, contudo os resultados obtidos foram considerados bons, encerrando-se portanto a fase de calibração dos controladores.

<i>string</i>	<i>valor decimal</i>	<i>parâmetro</i>
000	0	-10,0
001	1	-0,5
010	2	.3,0
011	3	.5,0
100	4	1,0
101	5	1,5
110	6	3,0
111	7	10,0

Tabela 5.2. São mostradas as strings possíveis, seus respectivos códigos decodificadores dados em valores decimais, bem como o valor, dado em decimal, associado aos parâmetros difusos dos controladores.

Uma vez determinadas as alternativas, poder-se-ia tentar verificar todas as possibilidades, isto é, executar todas as simulações necessárias para verificação do funcionamento do sistema com os parâmetros descritos. Por se tratar explicitamente de um problema de combinação, seriam necessárias:

Aplicação do Modelo Proposto

$$\begin{aligned}C_{24,8} &= 24 ! / (24-8)! \cdot 8! \\ &= 735471 \text{ simulações}\end{aligned}$$

Lavando-se em consideração que cada simulação consumiu em média 35 segundos em um microcomputador 486 DX2 66, com 16 MBytes de memória RAM, seriam necessários praticamente 298 dias para se obter o resultado final. Isto inviabiliza sua aplicação prática. Utilizando-se a otimização via GAs, através do teste de 360 alternativas, foram consumidas 2,1 horas de simulação, ou seja $489,5 \cdot 10^{-6}$ vezes menos que o necessário para verificar todas as possibilidades.

Analisando os resultados, constata-se que foram obtidas em todos os procedimentos de calibração uma convergência de valores. Contudo, pode ter havido uma convergência prematura, sobretudo na calibração do controlador de transporte, no problema de minimização do tempo máximo em atraso. De qualquer forma, os valores obtidos foram significativamente melhores que os valores alcançados pela aplicação pura das regras propostas.

5.6. Resultado da Simulação

Os dados obtidos a partir da simulação dos indivíduos em cada geração mostrou sempre uma melhora de desempenho do sistema ao longo do tempo, isto é, houve uma evolução das populações a cada geração. Nas figuras 5.2 a 5.5, são plotados os valores de *fitness* da população em termos de valores médios e máximos para a calibração dos controladores de fila e de transporte. As tabelas 5.3.e 5.4 indicam os valores nominais do *fitness* das populações em termos médios, máximos e mínimos.

Alguns comentários podem ser feitos com relação aos resultados. Observou-se em todos que o valor máximo era

Aplicação do Modelo Proposto

alcançado com relativa rapidez. Isto significa que talvez o ponto máximo de desempenho do sistema não tenha sido alcançado. A limitação da população contribui decisivamente para isto. Um maior número de indivíduos em cada geração possibilita uma maior ação exploratória, e por conseguinte, uma melhor chance de melhoria de desempenho. A diminuição da probabilidade de ocorrência de *crossover* também ajuda a aumentar a chance de melhoria de desempenho.

Também pode-se analisar individualmente as calibrações dos controladores. Na figura 5.2 (calibração do controlador de fila para minimização do atraso médio) observa-se a evolução das alternativas testadas. Apesar do máximo ter sido obtido já na primeira geração, a média do *fitness* da população cresceu ao longo do tempo até que houvesse convergência de todas as alternativas no ponto máximo.

Na figura 5.3 (calibração do controlador de fila para minimização do atraso máximo) observa-se que a melhor resposta foi obtida após algumas gerações, enquanto a média evoluiu constantemente. Nota-se, contudo, que houve uma geração onde o valor máximo de *fitness* decresceu. Isto pode ser considerado normal e transitório, pois o valor voltou a crescer na geração seguinte.

Para a calibração do controlador de transporte para minimização do atraso médio (figura 5.4), o mesmo fenômeno ocorreu, isto é, houve um ponto onde o valor máximo decresceu. Mas como pode-se observar continuou a haver a evolução tanto em termos de valor máximo quanto médios, significando que estava havendo uma convergência.

Finalmente para a figura 5.5 cabe ressaltar que o ponto de máximo desempenho foi alcançado e depois perdido. Isto significa que possivelmente ocorreu uma convergência

Aplicação do Modelo Proposto

prematura. Neste caso o analista deveria tentar fazer algumas alterações nos parâmetros iniciais dos GAs, tais como: aumentar população ou diminuir a probabilidade de ocorrência de *crossover*. Contudo, o valor máximo obtido ainda poderia ser considerado satisfatório.

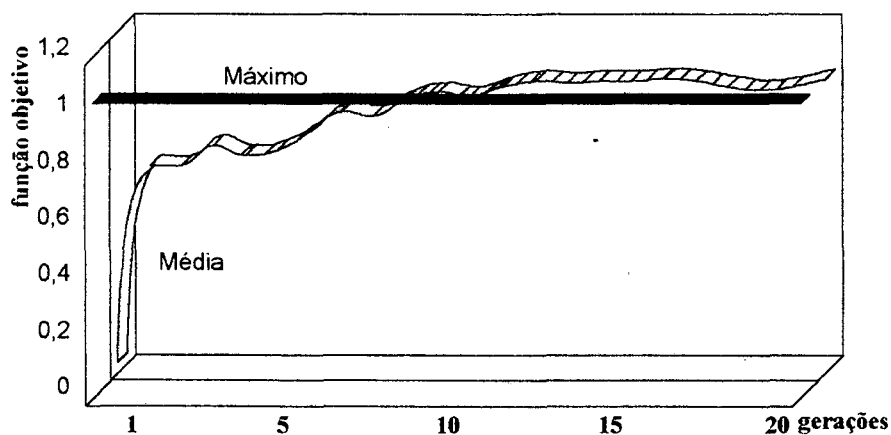


Figura 5.2. Evolução da população em termos do fitness, avaliados por valores médios e máximos durante a calibração do controlador de fila para condição de minimização do tempo médio em atraso.

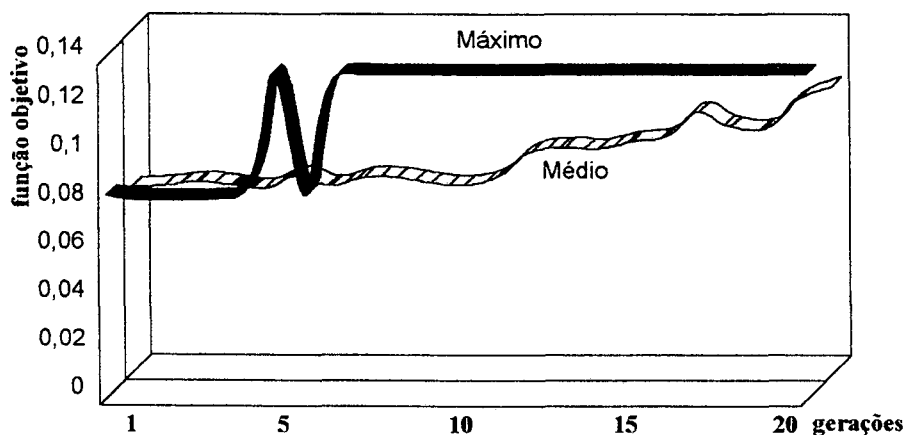


Figura 5.3. Evolução da população em termos do fitness, avaliados por valores médios e máximos durante a calibração do controlador de fila para condição de minimização do tempo máximo em atraso.

Aplicação do Modelo Proposto

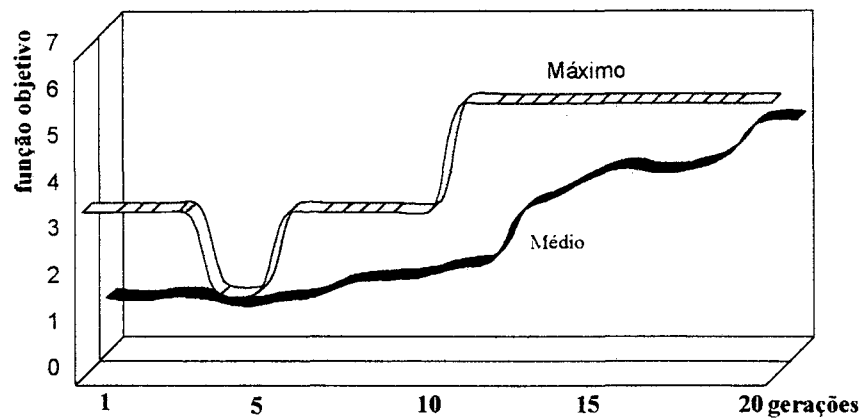


Figura 5.4. Evolução da população em termos do fitness, avaliados por valores médios e máximos durante a calibração do controlador de transporte para condição de minimização do tempo médio em atraso.

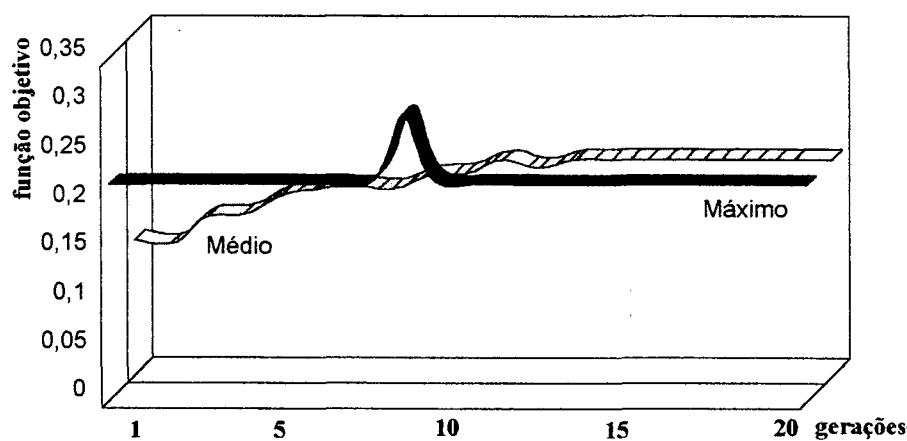


Figura 5.5. Evolução da população em termos do fitness, avaliados por valores médios e máximos durante a calibração do controlador de transporte para condição de minimização do tempo máximo em atraso.

Aplicação do Modelo Proposto

Otimização 1			Otimização 2		
Máx.	Mín.	Medida	Máx.	Mín.	Medida
1,0372	0,6085	0,7454	0,0833	0,0633	0,0763
1,0372	0,6085	0,7274	0,0817	0,0661	0,0772
1,0372	0,6085	0,7299	0,0817	0,0666	0,0786
1,0372	0,6091	0,7277	0,0820	0,0581	0,0774
1,0372	0,6628	0,8019	0,0834	0,0581	0,0761
1,0372	0,5815	0,7673	0,1331	0,0710	0,0812
1,0372	0,6353	0,7871	0,0834	0,0668	0,0772
1,0372	0,6657	0,8499	0,1331	0,0668	0,0807
1,0372	0,6715	0,9216	0,1331	0,0668	0,0806
1,0372	0,6443	0,9076	0,1331	0,0668	0,0784
1,0372	0,6696	0,9660	0,1331	0,0668	0,0776
1,0372	0,6696	0,9905	0,1331	0,0668	0,0812
1,0372	0,6757	0,9775	0,1331	0,0658	0,0914
1,0372	0,9176	1,0106	0,1331	0,0668	0,0928
1,0372	0,9176	1,0306	0,1331	0,0668	0,0925
1,0372	0,9176	1,0239	0,1331	0,0668	0,0954
1,0372	0,9176	1,0306	0,1331	0,0668	0,0973
1,0372	0,9176	1,0306	0,1331	0,0668	0,1076
1,0372	1,0372	1,0372	0,1331	0,0668	0,1015
1,0372	1,0372	1,0372	0,1331	0,0668	0,1008
$f(x) = (1. 10^6)/(x+1000)^2$			$f(x) = (1. 10^6)/(x)^2$		

Tabela 5.3. Resultados da simulação. As colunas relacionadas a otimização 1 são relativas a calibração do controlador de fila para condição de minimização do tempo médio de atraso. As colunas relacionadas a otimização 2 relacionam-se a minimização do tempo máximo de atraso. Os valores são dados em termos médios, mínimos e máximos. Por último são mostradas as funções objetivos utilizadas.

Aplicação do Modelo Proposto

Otimização 3			Otimização 4		
Max	Min	Média	Max	Min	Média
3,5769	0,7724	1,2113	0,2215	0,0736	0,1387
3,5769	0,7724	1,1523	0,2221	0,0765	0,1351
3,5769	0,6078	1,2323	0,2221	0,0863	0,1628
3,5769	0,6707	1,1745	0,2221	0,0863	0,1657
1,7967	0,7628	1,0236	0,2221	0,0837	0,1841
1,7967	0,6983	1,1448	0,2221	0,0746	0,1915
3,5769	0,6983	1,2322	0,2221	0,0756	0,1940
3,5769	0,7777	1,5237	0,2221	0,0755	0,1917
3,5769	0,8211	1,6010	0,2935	0,0755	0,2047
3,5769	1,0064	1,6534	0,2221	0,0755	0,2078
3,5769	1,0317	1,8420	0,2221	0,2068	0,2207
5,9078	1,0226	1,9654	0,2221	0,0756	0,2134
5,9078	0,8211	3,0828	0,2215	0,2215	0,2215
5,9078	0,8211	3,4087	0,2215	0,2215	0,2215
5,9078	1,0129	3,7793	0,2215	0,2215	0,2215
5,9078	1,1828	3,9847	0,2215	0,2215	0,2215
5,9078	1,1828	3,8975	0,2215	0,2215	0,2215
5,9078	1,5305	3,9812	0,2215	0,2215	0,2215
5,9078	1,5960	4,2667	0,2215	0,2215	0,2215
5,9078	1,1828	3,6809	0,2215	0,2215	0,2215
$f(x) = (1. 10^6)/(x+1000)^2$			$f(x) = (1. 10^6)/(x)^2$		

Tabela 5.4. Resultados da simulação. As colunas relacionadas a otimização 3 são relativas a calibração do controlador de transporte para condição de minimização do tempo médio de atraso. As colunas relacionadas a otimização 4 relacionam-se a minimização do tempo máximo de atraso. Os valores são dados em termos médios, mínimos e máximos. Por último são mostradas as funções objetivos utilizadas.

Aplicação do Modelo Proposto

Os valores obtidos para a calibração dos controladores são mostrados na tabela 5.5 e 5.6. As curvas de resposta dos mesmos são plotadas nas figuras 5.6 a 5.9

otimização	MIP		EDD		MNEP		FIFO	
	min	max	min	max	min	max	min	max
1	0,5	1,5	0,3	1,5	-10,0	3,0	1,0	3,0
2	-10,0	-0,5	-0,5	0,3	-10,0	3,0	1,5	3,0

Tabela 5.5. Valores obtidos para os parâmetros difusos após a calibração do controlador de fila. As linhas se relacionam respectivamente a cada otimização.

otimização	EDD		MCTED		MNPES		FIFO	
	min	max	min	max	min	max	min	max
3	0,5	3,0	0,3	1,0	-10,0	10,0	1,5	10,0
4	1,5	1,5	-10,0	10,0	1,0	1,0	1,5	3,0

Tabela 5.6. Valores obtidos para os parâmetros difusos após a calibração do controlador de transporte. As linhas se relacionam respectivamente a cada otimização.

Aplicação do Modelo Proposto

Controlador de Fila

curva de resposta para minimizar atraso médio

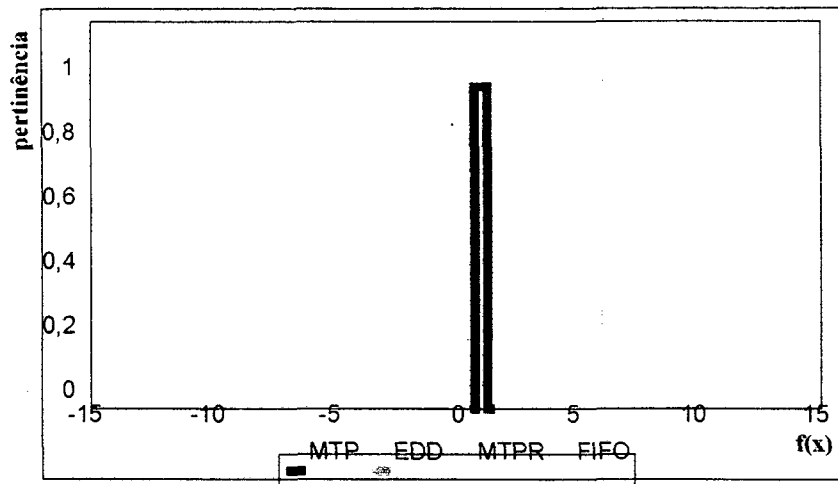


Figura 5.6. Curva de resposta para o controlador de fila calibrado para minimização do tempo médio em atraso.

Controlador de Fila

curva de resposta para minimizar atraso máximo

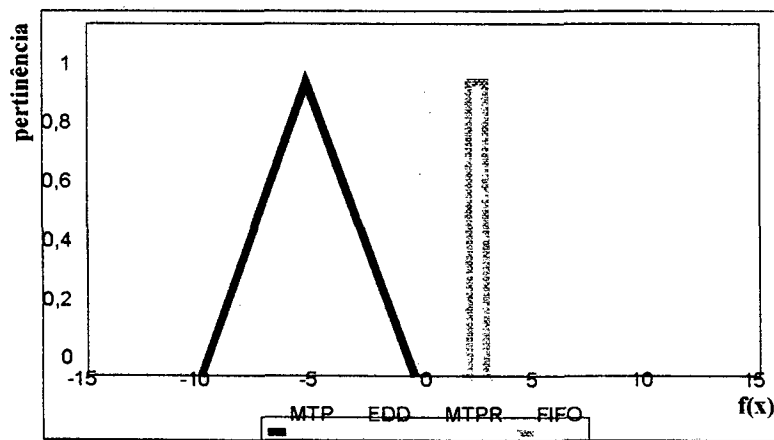


Figura 5.7. Curva de resposta para o controlador de fila calibrado para minimização do tempo máximo em atraso.

Aplicação do Modelo Proposto

Controlador de Transporte

curva de resposta para minimizar atraso médio

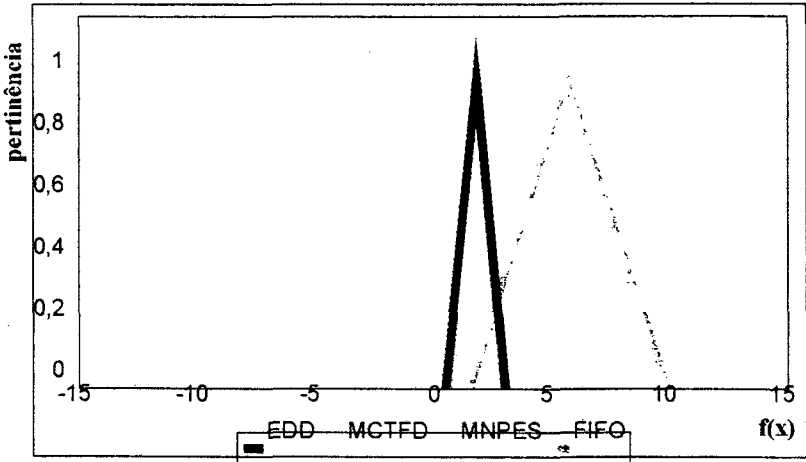


Figura 5.8. Curva de resposta para o controlador de transporte calibrado para minimização do tempo médio em atraso.

Controlador de Transporte

curva de resposta para minimizar máximo atraso

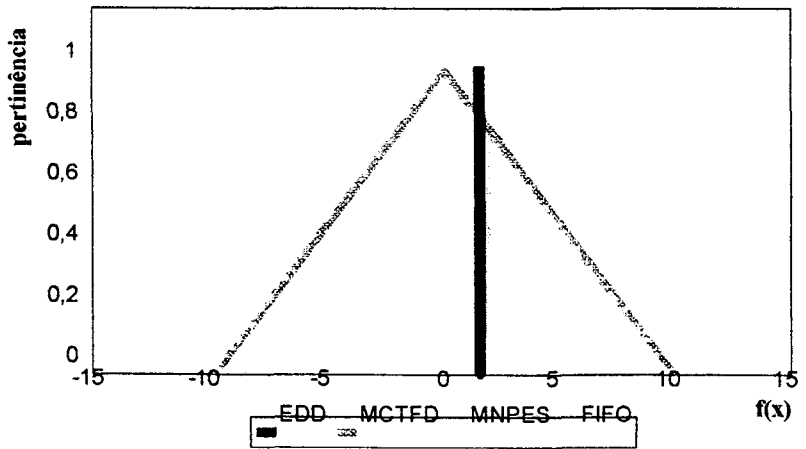


Figura 5.9. Curva de resposta para o controlador de transporte calibrado para minimização do tempo máximo em atraso.

Aplicação do Modelo Proposto

5.8. Comparação entre Modelo Proposto e Aplicação Pura das Regras

As Tabelas 5.6 e 5.7 e as figuras 5.10 a 5.14 indicam os ganhos conseguidos pelo emprego do método proposto em relação a aplicação pura de uma das regras. Isto foi feito através do isolamento de cada regra, isto é, aplicou-se apenas uma das regras do controlador em questão, mantendo-se o outro controlador calibrado. Observa-se que o modelo proposto sempre obteve resultados superiores. Algumas vezes o resultado se apresentou extremamente superior a alguma destas regras. Este fraco desempenho é devido a completa inadequação de tal regra a situação do sistema, como por exemplo as regras MCTFD, MNPES e FIFO para minimização do tempo médio em atraso aplicada à liberação de peças para transporte.

A dinâmica do sistema, com grande variação de demanda nas máquinas, é responsável pela dificuldade de aplicação de uma determinada regra à política operacional. A escolha dinâmica de regras, feita pelos controladores de fila e transporte, se mostrou mais adequada, bem como mais robusta. Isto leva a crer que os objetivos pretendidos foram alcançados.

	MIP	EDD	MIPR	FIFO
atraso médio (min.).	273,7	239,0	184,8	160,0
modelo (min.).	-18,1			
ganho (min.).	291,7	257,1	202,9	178,0
máximo atraso(min.).	3869,1	3948,6	3678,6	3507,2
modelo (min)	2759,7			
ganho (min.)	1109,4	738,9	918,9	747,5

Tabela 5.7. Atraso médio e máximo atraso para cada regra utilizada pelo controlador de fila. Os ganhos pela aplicação do modelo proposto são relativos aos seguintes valores: 2759,7min.(atrasado para atraso máximo) e -18,1min.(adiantado, para atraso médio).

Aplicação do Modelo Proposto

	EDD	MCTPD	MINFES	FIFO
atraso médio (min.)	-131,8	223,2	20,7	20,7
modelo (min.)	-588,5			
ganho (min.)	456,7	811,7	609,2	609,7
máximo atraso (min.)	3404,0	2124,7	2741,0	2741,0
modelo (min.)	1845,8			
ganho (min.)	1558,2	278,9	895,2	895,2

Tabela 5.8. Atraso médio e máximo atraso para cada regra utilizada pelo controlador de transporte. Os ganhos pela aplicação do modelo proposto são relativos aos seguintes valores: -588,5min. (adiantado para atraso médio) e 1845,8 (atrasado para atraso máximo).

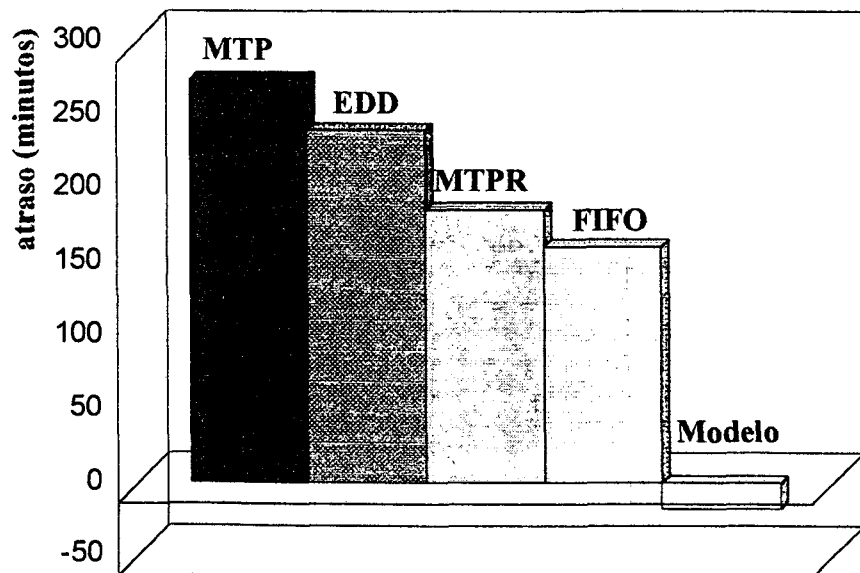


Figura 5.10. Gráfico de desempenho em termos de atraso médio para cada regra utilizada pelo controlador de fila e pela aplicação dinâmica das mesmas através do modelo proposto (os valores são dados em minutos).

Aplicação do Modelo Proposto

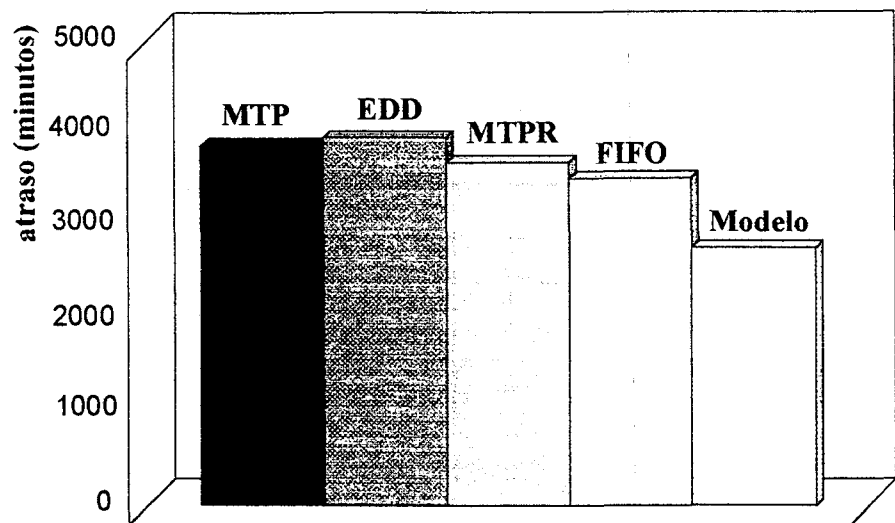


Figura 5.11. Gráfico de desempenho em termos de atraso máximo para cada regra utilizada pelo controlador de fila e pela aplicação dinâmica das mesmas através do modelo proposto (os valores são dados em minutos).

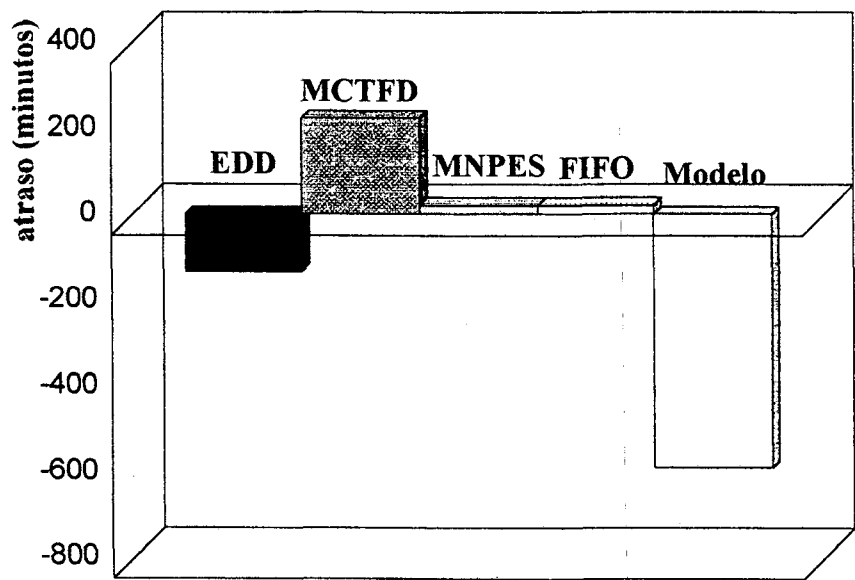


Figura 5.12. Gráfico de desempenho em termos de atraso médio para cada regra utilizada pelo controlador de transporte e pela aplicação dinâmica das mesmas através do modelo proposto (os valores são dados em minutos).

Aplicação do Modelo Proposto

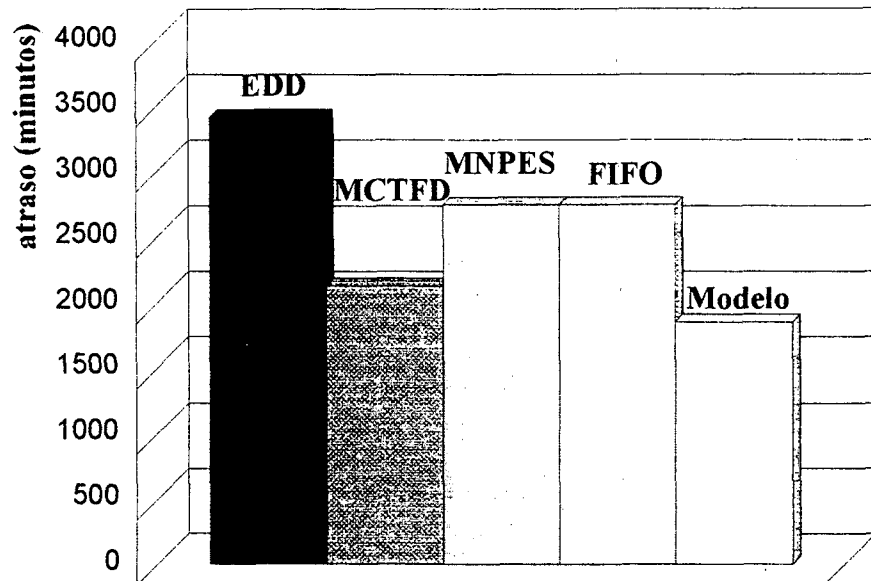


Figura 5.13. Gráfico de desempenho em termos de atraso médio para cada regra utilizada pelo controlador de fila e pela aplicação dinâmica das mesmas através do modelo proposto (os valores são dados em minutos).

A conclusão que se pode ter a partir dos dados conseguidos é que o modelo proposto funciona, e é aplicável ao caso em questão, obtendo-se resultados bastante satisfatórios.

5.9. Sumário

Neste capítulo foi descrito uma aplicação da modelo proposto para otimização da produção. Primeiramente foram explicitadas todas as características do modelo simulado, bem como considerações a respeito das mesmas. Em seguida descreveu-se o método de otimização feito através do uso das GAs. Finalmente foram apresentados os resultados oriundos da otimização, podendo-se constatar ganho apreciável sob todas as condições. Isto vem a comprovar a eficiência do modelo proposto, e portanto validar sua aplicação em problemas relacionados à otimização de processos de manufatura.

CONCLUSÕES E RECOMENDAÇÕES

Buscou-se neste trabalho desenvolver um modelo alternativo para geração de *scheduling*, utilizando para isto algumas das ferramentas mais modernas disponíveis atualmente. Este modelo proposto foi acompanhado de um exemplo aplicado em uma indústria de bens não seriados, combinando o uso de simulação, algoritmos genéticos e sistemas difusos.

Os experimentos realizados comprovam a utilidade de sua aplicação, tendo os resultados ganhos significativos em relação às estratégias usualmente utilizadas.

Neste capítulo faz-se uma síntese de tudo aquilo que foi explorado, estudado e concluído, apresentando os resultados obtidos e as contribuições consideradas relevantes. As limitações são também apresentadas de forma a incentivar novas pesquisas.

6.1. Resultados e Contribuições da Pesquisa Os sistemas produtivos operando sob encomenda estão sujeitos a uma grande variedade de produtos, o que aumenta de sobremaneira a complexidade do sistema. Sistemas de auxílio ao planejamento da produção, como por exemplo o MRP e MRPII, não levam em consideração problemas relacionados ao dia a dia da empresa, isto é, as decisões são baseadas em dados estáticos, não levando em consideração restrições oriundas do comportamento dinâmico do sistema, tais como as capacidades dos recursos, as interrelações entre máquinas e equipamentos, variações de tempos de processamento, etc. Portanto o

Conclusões e Recomendações

planejamento gerado por tais sistemas muitas vezes tendem a ser falhos ou longe de apresentar um desempenho que possa ser considerado ótimo. O método aqui proposto procura responder a algumas destas considerações sem contudo deixar de ser um método prático.

A principal contribuição deste trabalho foi a de possibilitar que a tomada de decisão fosse feita dinamicamente durante a simulação. Isto permitiu que ajustes externos fossem feitos, aumentando o desempenho do sistema, ao mesmo tempo que se manteve a flexibilidade do modelo de simulação. Sua aplicação conjunta com algoritmos de otimização veio a comprovar ainda seu enorme potencial de uso concomitante ao emprego de outras ferramentas.

É indubitável que o uso da simulação possa ser ampliado, saindo do estágio atual onde é utilizada apenas por grandes empresas em problemas muitas vezes relacionados apenas a projetos de *layout*. O exemplo mostrado foi desenvolvido em uma indústria de médio porte, demonstrando resultados marcantes. Isto indica que há possibilidade de ampliação do universo de usuários de tal tecnologia. Especificamente com relação a região sul do Brasil, onde existem diversas empresas com este perfil, este trabalho tem sua importância aumentada, contribuindo assim como alternativa de aumento da produtividade das empresas da região.

Um dos principais problemas relacionados a aplicação deste modelo se refere a necessidade de compra de um simulador de processos. Tais programas têm seus preços em um patamar bastante alto para as médias e pequenas empresas. Contudo é factível esperar que os preços destes softwares baixem consideravelmente em um futuro próximo, prática comum em mercados competitivos, como o da informática.

Conclusões e Recomendações

Os resultados obtidos, apesar de animadores, devem ser encarados com cautela. Apesar de acreditar que sempre haverá um aumento no desempenho do sistema, este seguramente variará de empresa para empresa. Acredita-se que quanto maior for a demanda e utilização das máquinas e equipamentos, maior será o potencial de uso deste modelo. Em indústrias operando com baixa utilização de seus recursos provavelmente não terão problemas relacionados a estoques de produtos semi-acabados ou estoques intermediários, que é basicamente o que o sistema procura resolver. Portanto, quanto maiores forem as filas esperando algum recurso maior será o ganho obtido e maior será o retorno conseguido.

Apesar de serem tratados apenas os problemas relacionados a atraso, a minimização de custos ou do *leadtime*, a maximização de lucros ou do volume de produção podem ser também facilmente tratados. Para isto basta incorporá-los ao modelo de simulação. Isto amplia o escopo de problemas alcançados por esta proposta.

6.2. Limitações do Trabalho

Como qualquer trabalho de pesquisa, a abrangência deste estudo é limitada. Seus limites foram definidos e apresentados ao longo do trabalho. Todos os resultados numéricos foram obtidos a partir de um sistema real, com dados de demanda e prazos fictícios. Desta forma, todas as conclusões derivadas dos experimentos são válidas apenas para aquele sistema e condição.

No que diz respeito ao modelo e ao sistema computacional desenvolvidos, estes são mais genéricos por sua própria natureza, podendo ser aplicados a outros sistemas semelhantes.

Os experimentos realizados consideraram algumas poucas estratégias de controle operacional. Existem muitas outras

Conclusões e Recomendações

regras passíveis de utilização que poderiam ser implementadas. Tais regras poderiam ser até mais apropriadas para algumas situações que as utilizadas neste trabalho.

6.3. Recomendações para Futuras Pesquisas

As pesquisas feitas neste trabalho sugerem algumas áreas relevantes para o desenvolvimento de outros estudos. Nesta seção aponta-se algumas destas, recomendando futuros trabalhos. Os itens abaixo relacionados não seguem nenhuma ordem de importância.

6.3.1. Desenvolvimento de novos controladores

Os controladores aqui apresentados limitaram-se apenas ao tratamento da questão do gerenciamento de filas. Isto pode ser considerado uma limitação, uma vez que o sistema não tem como determinar alterações na ordem de entrada da peça para fabricação quando não existir fila, isto é, mesmo que uma fila apareça um infinitésimo de tempo após uma peça ter dado entrada no processo não seria possível que algumas das peças que chegaram neste intervalo infinitesimal tomar o recurso antes daquela que já deu entrada. Isto é basicamente função do planejamento, que não foi tratado neste trabalho. Apesar disto, políticas de planejamento também poderiam ser tratadas através de controladores de despacho de ordens de fabricação, fazendo com que em determinadas situações peças esperem na fila mesmo que o recurso a jusante esteja livre.

Controladores de máquinas alternativas também seriam bastante úteis para gerenciar o despacho de peças para máquinas que pudessem fazer o mesmo processo. Isto simplificaria também a fase de planejamento, ao mesmo tempo que possibilitaria a incorporação de características dinâmicas a esta decisão.

Conclusões e Recomendações

6.3.2. Aplicação em Ambientes Tipo Flow-Shop

Estudou-se aqui apenas a aplicação em ambientes do tipo *job-shop*. Acredita-se que a migração para ambientes tipo *flow-shop* deva ser factível. As características do tipo tamanho de lote, velocidade de *conveyors*, mão de obra disponível não teriam dificuldade em serem tratadas pelo simulador.

6.3.3. Aplicação de Mais Regras nos Controladores

Tratou-se aqui apenas da aplicação de quatro regras para cada um dos controladores. A utilização de mais regras aumentaria a possibilidade de melhoria no desempenho do sistema, uma vez que mais situações seriam cobertas com regras mais específicas.

6.3.4. Aperfeiçoamento do Controlador Difuso

O método de seleção escolhido, através da escolha da regra de maior pertinência, possui alguns inconvenientes. KOSCO [1991] analisa especificamente este assunto e sugere o método de cálculo através da determinação da centróide. Sugere-se, a partir deste estudo, que sejam testadas a aplicação deste, ou de outros métodos, visando aumentar o desempenho dos controladores difusos, e portando do próprio sistema, ao mesmo tempo que possibilite o efetivo uso das regras FAMs.

LISTAGEM DO MODELO DE SIMULAÇÃO

```

25$      CREATE,      1,50000:,1;
18$      TRACE,      -1,"-Entity Created\n";
21$      ASSIGN:      Picture=Default:NEXT(1$);

1$      TRACE,      -1,"-Writing to FUNCAO.DAT\n";
26$      WRITE,      FUNCAO.DAT:TMAX(ATRASO):NEXT(0$);

0$      TRACE,      -1,"-Disposing entity\n";
27$      DISPOSE;
30$      CREATE,      1,0:,1:NEXT(28$);

28$      TRACE,      -1,"-Reading from parametr.dat \n";
31$      READ,
parametr.dat:a1,a2,a3,a4,a5,a6,a7,a8:NEXT(29$);

29$      TRACE,      -1,"-Disposing entity\n";
32$      DISPOSE;
13$      STATION,      Saida;
33$      TRACE,      -1,"-Arrived to station Saida\n";
34$      DELAY:      0.000:NEXT(12$);

12$      TRACE,      -1,"-Updating Tally atraso \n";
37$      TALLY:      atraso,Interval(Tentrega),1:NEXT(17$);

17$      STATION,      fim;
45$      TRACE,      -1,"-Arrived to station fim\n";
75$      DELAY:      0.;
53$      COUNT:      fim_C,1;
48$      TALLY:      FlowTime,Interval(TENTRADA),1;
38$      TRACE,      -1,"-Disposing entity\n";
46$      DISPOSE;
100$      CREATE,      1:100:MARK(TENTRADA);
99$      ASSIGN:      NS=DISC(.5,1,1,2):
Tentrega=TNOW+400:
Trestante=300;

140$      STATION,      Entrada;
91$      TRACE,      -1,"-Arrived to system at station
Entrada\n";
137$      ASSIGN:      Picture=Default;
116$      DELAY:      0.;
84$      TRACE,      -1,"-Transferred to next station in
sequence\n";
107$      ROUTE:      0.,SEQ;
2$      TRACE,      -1,"-Making assignments\n";
161$      ASSIGN:      Funcao=(Tentrega-
Tnow)/(Trestante):NEXT(160$);

160$      TRACE,      -1,"-Duplicating entities\n";
162$      DUPLICATE:      1,156$:
1,153$:
1,150$:NEXT(159$);

159$      TRACE,      -1,"-Choosing from 4 options\n";
163$      BRANCH,      1:If, Funcao<MIN(a1,a2),158$,Yes:
If, Funcao<(a1+a2)/2,157$,Yes:
If, Funcao<MAX(a1,a2),141$,Yes:
Else,158$,Yes;

158$      TRACE,      -1,"-Making assignments\n";
164$      ASSIGN:      PERTINENCIA=0:
R_SELECTED=0:NEXT(145$);

145$      TRACE,      -1,"-Waiting for batch size of 4\n";
166$      QUEUE,      fila_REGRA;
174$      COMBINE,      :4,First;

```

168\$	TRACE,	-1,"-Formed a permanent batch of size
4\n";		
173\$	DELAY:	0.000:NEXT(144\$);
144\$	TRACE,	-1,"-Choosing from 3 options\n";
176\$	BRANCH,	1:If,PLC==CONTROLADOR 1,9\$,Yes:
		If,PLC==CONTROLADOR 2,10\$,Yes:
		If,PLC==CONTROLADOR 3,11\$,Yes:
157\$	TRACE,	-1,"-Making assignments\n";
177\$	ASSIGN:	PERTINENCIA=(Funcao-MIN(a1,a2))/ABS(a2-
a1):		R_SELECTED=0:NEXT(145\$);
141\$	TRACE,	-1,"-Making assignments\n";
178\$	ASSIGN:	PERTINENCIA=2/(MIN(a1,a2)-
MAX(a1,a2))*(Funcao-(a1+a2)/2)+1:		R_SELECTED=0:NEXT(145\$);
156\$	TRACE,	-1,"-Choosing from 4 options\n";
179\$	BRANCH,	1:If,Funcao<MIN(a3,a4),155\$,Yes:
		If,Funcao<(a3+a4)/2,154\$,Yes:
		If,Funcao<MAX(a3,a4),142\$,Yes:
		Else,155\$,Yes:
155\$	TRACE,	-1,"-Making assignments\n";
180\$	ASSIGN:	PERTINENCIA=0:
		R_SELECTED=1:NEXT(145\$);
154\$	TRACE,	-1,"-Making assignments\n";
181\$	ASSIGN:	PERTINENCIA=(Funcao-MIN(a3,a4))/ABS(a4-
a3):		R_SELECTED=1:NEXT(145\$);
142\$	TRACE,	-1,"-Making assignments\n";
182\$	ASSIGN:	PERTINENCIA=2/(MIN(a3,a4)-
MAX(a3,a4))*(Funcao-(a3+a4)/2)+1:		R_SELECTED=1:NEXT(145\$);
153\$	TRACE,	-1,"-Choosing from 4 options\n";
183\$	BRANCH,	1:If,Funcao<MIN(a5,a6),152\$,Yes:
		If,Funcao<(a5+a6)/2,151\$,Yes:
		If,Funcao<MAX(a5,a6),143\$,Yes:
		Else,152\$,Yes:
152\$	TRACE,	-1,"-Making assignments\n";
184\$	ASSIGN:	PERTINENCIA=0:
		R_SELECTED=2:NEXT(145\$);
151\$	TRACE,	-1,"-Making assignments\n";
185\$	ASSIGN:	PERTINENCIA=(Funcao-MIN(a5,a6))/ABS(a6-
a5):		R_SELECTED=2:NEXT(145\$);
143\$	TRACE,	-1,"-Making assignments\n";
186\$	ASSIGN:	PERTINENCIA=2/(MIN(a5,a6)-
MAX(a5,a6))*(Funcao-(a5+a6)/2)+1:		R_SELECTED=2:NEXT(145\$);
150\$	TRACE,	-1,"-Choosing from 4 options\n";
187\$	BRANCH,	1:If,Funcao<MIN(a7,a8),149\$,Yes:
		If,Funcao<(a7+a8)/2,148\$,Yes:
		If,Funcao<MAX(a7,a8),147\$,Yes:
		Else,149\$,Yes:
149\$	TRACE,	-1,"-Making assignments\n";
188\$	ASSIGN:	PERTINENCIA=0:
		R_SELECTED=3:NEXT(145\$);
148\$	TRACE,	-1,"-Making assignments\n";
189\$	ASSIGN:	PERTINENCIA=(Funcao-MIN(a7,a8))/ABS(a7-
a8):		R_SELECTED=3:NEXT(145\$);
147\$	TRACE,	-1,"-Making assignments\n";
190\$	ASSIGN:	PERTINENCIA=2/(MIN(a7,a8)-
MAX(a7,a8))*(Funcao-(a7+a8)/2)+1:		


```

R_SELECTED=3:NEXT(145$);

9$      TRACE,      -1,"-Making assignments\n";
191$    ASSIGN:      REGRA_1=R_SELECTED:NEXT(8$);

8$      TRACE,      -1,"-Choosing from 4 options\n";
208$    BRANCH,      1:If,R_SELECTED==0,203$,Yes:
                        If,R_SELECTED==1,202$,Yes:
                        If,R_SELECTED==2,198$,Yes:
                        Else,194$,Yes:
203$    SEARCH,      Maquina
1_Q,1,NQ:MIN(TProcesso):NEXT(201$);

201$    TRACE,      -1,"-Choosing from 2 options\n";
209$    BRANCH,      1:If,j>1,199$,Yes:
                        Else,195$,Yes:
199$    REMOVE:      j,Maquina 1_Q,200$:NEXT(195$);

195$    TRACE,      -1,"-Disposing entity\n";
210$    DISPOSE;
200$    INSERT:      207$,1;
207$    QUEUE,      Maquina 1_Q;
206$    SEIZE,      1:Maquina 1,1:NEXT(197$);

197$    TRACE,      -1,"-Duplicating entities\n";
211$    DUPLICATE:   1,196$:NEXT(205$);

205$    DELAY:      TProcesso:NEXT(192$);

192$    TRACE,      -1,"-Making assignments\n";
212$    ASSIGN:      Maquina 1_CIQ=Maquina 1_CIQ-TProcesso:
Trestante=Trestante-
Tprocesso:NEXT(204$);

204$    RELEASE:     Maquina 1,1:NEXT(14$);

196$    TRACE,      -1,"-Making assignments\n";
213$    ASSIGN:      Maquina 1_NIQ=NQ(Maquina 1_Q):
Trestante=Trestante - TProcesso:
PLC=CONTROLADOR 1:NEXT(2$);

202$    SEARCH,      Maquina
1_Q,1,NQ:MIN(Tentrega):NEXT(201$);

198$    SEARCH,      Maquina 1_Q,1,NQ:MIN(SLACK):NEXT(201$);

194$    SEARCH,      Maquina 1_Q,1,NQ:MIN(ARRIVAL
TIME):NEXT(201$);

7$      STATION,     Maquina 1_S;
220$    TRACE,      -1,"-Arrived to station Maquina 1_S\n";
240$    STORE:      Maquina 1_S_S1;
241$    DELAY:      0.;
226$    UNSTORE:    ;
221$    DELAY:      0.000:NEXT(193$);

193$    TRACE,      -1,"-Making assignments\n";
242$    ASSIGN:      Maquina 1_CIQ=Maquina 1_CIQ+Tprocesso:
ARRIVAL TIME=TNOW:NEXT(207$);

14$     DELAY:      0.00;
273$    DELAY:      0.;
247$    TRACE,      -1,"-Transferred to next station in
sequence\n";
263$    ROUTE:      0.,SEQ;
10$     TRACE,      -1,"-Making assignments\n";
285$    ASSIGN:      REGRA_2=R_SELECTED:NEXT(6$);

6$      TRACE,      -1,"-Choosing from 4 options\n";
302$    BRANCH,      1:If,R_SELECTED==0,297$,Yes:
                        If,R_SELECTED==1,296$,Yes:
                        If,R_SELECTED==2,292$,Yes:
                        Else,288$,Yes:

```

```

297$          SEARCH,          Maquina
2_Q,1,NQ:MIN(TProcesso):NEXT(295$);

295$          TRACE,          -1,"-Choosing from 2 options\n";
303$          BRANCH,          1:If,j>1,293$,Yes:
                               Else,289$,Yes;
293$          REMOVE:          j,Maquina 2_Q,294$:NEXT(289$);

289$          TRACE,          -1,"-Disposing entity\n";
304$          DISPOSE;
294$          INSERT:          301$,1;
301$          QUEUE,          Maquina 2_Q;
300$          SEIZE,          1:Maquina 2,1:NEXT(291$);

291$          TRACE,          -1,"-Duplicating entities\n";
305$          DUPLICATE:        1,290$:NEXT(299$);

299$          DELAY:          TProcesso:NEXT(286$);

286$          TRACE,          -1,"-Making assignments\n";
306$          ASSIGN:          Maquina 2_CIQ=Maquina 2_CIQ-TProcesso:
                               Trestante=Trestante-
Tprocesso:NEXT(298$);

298$          RELEASE:          Maquina 2,1:NEXT(15$);

290$          TRACE,          -1,"-Making assignments\n";
307$          ASSIGN:          Maquina 2_NIQ=NQ(Maquina 2_Q):
                               Trestante=Trestante - TProcesso:
                               PLC=CONTROLADOR 2:NEXT(2$);

296$          SEARCH,          Maquina
2_Q,1,NQ:MIN(Tentrega):NEXT(295$);

292$          SEARCH,          Maquina 2_Q,1,NQ:MIN(SLACK):NEXT(295$);

288$          SEARCH,          Maquina 2_Q,1,NQ:MIN(ARRIVAL
TIME):NEXT(295$);

5$            STATION,          Maquina 2_S;
314$          TRACE,          -1,"-Arrived to station Maquina 2_S\n";
334$          STORE:          Maquina 2_S_S1;
335$          DELAY:          0.;
320$          UNSTORE:         ;
315$          DELAY:          0.000:NEXT(287$);

287$          TRACE,          -1,"-Making assignments\n";
336$          ASSIGN:          Maquina 2_CIQ=Maquina 2_CIQ+Tprocesso:
                               ARRIVAL TIME=TNOW:NEXT(301$);

15$           DELAY:          0.00;
367$           DELAY:          0.;
341$           TRACE,          -1,"-Transferred to next station in
sequence\n";
357$           ROUTE:          0.,SEQ;
11$           TRACE,          -1,"-Making assignments\n";
379$           ASSIGN:          REGRA_3=R_SELECTED:NEXT(4$);

4$            TRACE,          -1,"-Choosing from 4 options\n";
396$           BRANCH,          1:If,R_SELECTED==0,391$,Yes:
                               If,R_SELECTED==1,390$,Yes:
                               If,R_SELECTED==2,386$,Yes:
                               Else,382$,Yes;
391$           SEARCH,          Maquina
3_Q,1,NQ:MIN(TProcesso):NEXT(389$);

389$           TRACE,          -1,"-Choosing from 2 options\n";
397$           BRANCH,          1:If,j>1,387$,Yes:
                               Else,383$,Yes;
387$           REMOVE:          j,Maquina 3_Q,388$:NEXT(383$);

383$           TRACE,          -1,"-Disposing entity\n";
398$           DISPOSE;

```

```

388$      INSERT:      395$,1;
395$      QUEUE,      Maquina 3_Q;
394$      SEIZE,      1:Maquina 3,1:NEXT(385$);

385$      TRACE,      -1,"-Duplicating entities\n";
399$      DUPLICATE:   1,384$:NEXT(393$);

393$      DELAY:      TProcesso:NEXT(380$);

380$      TRACE,      -1,"-Making assignments\n";
400$      ASSIGN:      Maquina 3_CIQ=Maquina 3_CIQ-TProcesso:
Trestante=Trestante-
Tprocesso:NEXT(392$);

392$      RELEASE:     Maquina 3,1:NEXT(16$);

384$      TRACE,      -1,"-Making assignments\n";
401$      ASSIGN:      Maquina 3_NIQ=NQ(Maquina 3_Q):
Trestante=Trestante - TProcesso:
PLC=CONTROLADOR 3:NEXT(2$);

390$      SEARCH,      Maquina
3_Q,1,NQ:MIN(Tentrega):NEXT(389$);

386$      SEARCH,      Maquina 3_Q,1,NQ:MIN(SLACK):NEXT(389$);

382$      SEARCH,      Maquina 3_Q,1,NQ:MIN(ARRIVAL
TIME):NEXT(389$);

3$        STATION,      Maquina 3_S;
408$      TRACE,      -1,"-Arrived to station Maquina 3_S\n";
428$      STORE:      Maquina 3_S_S1;
429$      DELAY:      0.;
414$      UNSTORE:     ;
409$      DELAY:      0.000:NEXT(381$);

381$      TRACE,      -1,"-Making assignments\n";
430$      ASSIGN:      Maquina 3_CIQ=Maquina 3_CIQ+Tprocesso:
ARRIVAL TIME=TNOW:NEXT(395$);

16$       DELAY:      0.00;
461$      DELAY:      0.;
435$      TRACE,      -1,"-Transferred to next station in
sequence\n";
451$      ROUTE:      0.,SEQ;

```

```

PROJECT,      TESE,Fernando M Loureiro;

ATTRIBUTES:   TProcesso:
               Tentrega:
               Trestante:
               TENTRADA:
               ARRIVAL TIME:
               PLC:
               NUMBER IN QUEUE:
               CHARGE:
               R_SELECTED:
               SLACK:
               PERTINENCIA:
               Funcao;

FILES:         parametr.dat,"parametr.dat",(),Free Format,Dispose:
               FUNCAO.DAT,"FUNCAO.DAT",(),Free Format;

STORAGES:      Maquina 1_S_S1:
               Maquina 2_S_S1:
               Maquina 3_S_S1;

VARIABLES:     REGRA_1:
               REGRA_2:
               REGRA_3:
               Maquina 1_CIQ:
               Maquina 1_NIQ:
               Maquina 2_CIQ:
               Maquina 2_NIQ:
               Maquina 3_CIQ:
               Maquina 3_NIQ:
               a1:
               a2:
               a3:
               a4:
               a5:
               a6:
               a7:
               a8;

QUEUES:        Maquina 1_Q:
               Maquina 2_Q:
               Maquina 3_Q:
               fila_REGRA,HVF(Pertinencia);

PICTURES:      Default;

RESOURCES:     Maquina 1:
               Maquina 2:
               Maquina 3;

STATIONS:      Maquina 1_S:
               MAQUINA 2_S:
               MAQUINA 3_S:
               Entrada:
               Saida:
               fim:
               CONTROLADOR 1:
               CONTROLADOR 2:
               CONTROLADOR 3;

SEQUENCES:     Peca 1,MAQUINA 1_S,,,TProcesso=NORM(180,40)&MAQUINA
2_S,,,TProcesso=Normal(60,40)&MAQUINA 3_S,,,TProcesso=
               Normal(67,20)&Saida:
               Peca 2,MAQUINA 3_S,,,TProcesso=NORM(80,34)&MAQUINA
2_S,,,TProcesso=NORM(130,34)&MAQUINA 3_S,,,TProcesso=
               NORM(40,45)&Saida;

COUNTERS:      fim_C;

TALLIES:       FlowTime:
               atraso;

```

DSTATS: NQ(fila_REGRA),# in fila_REGRA;
REPLICATE, 1,0.0,50000,Yes,Yes;

ANEXO II

LISTAGEM DAS ROTINAS DOS ALGORITMOS GENÉTICOS

```
/*
  Algoritmo Genético
  David Goldberg
  Modificado por Fernando M. Loureiro
  26/05/94
*/
#include "ga.h"
#include <process.h>
#include <windows.h>
// Global Variables
population      *oldpop, *newpop; // 2 nonoverlapping
populations
int              popsize, lchrom, gen, maxgen;
double          pcross, pmutation, sumfitness;
int             nmutation, ncross;           // Int statistics
double          avg, max, min;              // Real
statistics
FILE             *arquivo, *arquivo2, *arquivo3;
void main(void)
{
  int i,j;
  char *apt_tmp;
  gen = 0;
  if (!(oldpop = new population[MAXPOP]))
  {
    printf("Insufficient memory for name");
    exit (1);
  }
  if (!(newpop = new population[MAXPOP]))
  {
    printf("Insufficient memory for name");
    exit (1);
  }
  for(i=0; i<=MAXPOP; i++)
  {
    newpop[i] = new individual;
    oldpop[i] = new individual;
    // Analisar se , necessario
    for (j=0; j<=MAXSTRING; j++)
    {
      oldpop[i]->chrom[j] = ' ';
      newpop[i]->chrom[j] = ' ';
    }
  }
  apt_tmp = new char;
  Initilize();
  printf ( " MAX          AVG \n" );
  do
  {
    gen++;
    Generation();
    Statistic( popsize, &max, &avg, &min, &sumfitness, newpop );
    Report( gen, arquivo, arquivo2 );
    printf ( " %5.3f %5.3f \n", max, avg );
    apt_tmp = (char *)oldpop;
    oldpop = newpop;
    newpop = (population *)apt_tmp;
  }
  while ( gen < maxgen);
}
```

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <conio.h>
#include "ga.h"
#include "random.h"
#include <process.h>
// Global Variables
extern population *oldpop, *newpop; // 2
nonoverlapping populations
extern int popsize, lchrom, gen, maxgen;
extern double pcross, pmutation, sumfitness;
extern int nmutation, ncross; // Int
statistics
extern double avg, max, min; // Real
statistics
extern FILE *arquivo, *arquivo2, *arquivo3;
double COEF_NORM; // Coefficient to normalize objective
function
// Objective Function
double Objfunc(double x)
{
    const n = 10;
    int j, para1, para2, para3, parametro;
    double powerof2;
    float parsaida, tabela[16], i;
    powerof2 = 1.0;
    tabela[1] = -.5; tabela[2] = .3; tabela[3] = .5; tabela[4] = 1.0;
    tabela[5] = 1.5;
    tabela[6] = 3; tabela[7] = 99; tabela[0] = -99;
    arquivo3 = fopen("parametr.dat", "w");
    for (j=1; j<lchrom; j++)
        powerof2 = powerof2 * 2.0;
    for (j=lchrom; j>=1; j=j-3)
    {
        para1=0; para2=0; para3=0;
        if (x>powerof2)
        {
            (para1=1);
            x -= powerof2;
        }
        powerof2 = powerof2/2;
        if (x>powerof2)
        {
            (para2=1);
            x -= powerof2;
        }
        powerof2 = powerof2/2;
        if (x>powerof2)
        {
            (para3=1);
            x -= powerof2;
        }
        powerof2 = powerof2/2;
        printf("bit %i %i %i -", para1, para2, para3);
        parametro = (para1*4+para2*2+para3);
        parsaida = tabela[parametro];
        printf ("parametro de saida: %f\n", parsaida);
        fprintf (arquivo3, "%f\n", parsaida);
    }
    fclose(arquivo3);
    system("siman teste1.p");
    arquivo = fopen("funcao.dat", "r");
    fscanf(arquivo, "%f", &i);
    i = 1000000/(i*i);
    printf ("o valor lido foi: %f\n", i);
    fclose(arquivo);
    return (i);
}
// Decode string as unsigned integer true = 1 and false = 0
double Decode( chromosome chrom, int lbits)
{
    int j;
    double accum, powerof2;

```

```

    accum = 0.0;
    powerof2 = 1.0;
    for (j=1; j<= lbits; j++)
    {
        if ( chrom[j] )
            accum += powerof2;
        powerof2 = powerof2 * 2.0;
    }
    return accum;
}

void Statistic( int popsize, double *max, double *avg, double *min,
               double *sumfitness, population *pop)
{
    int j;
    // Initailiaze
    *sumfitness = pop[1]->fitness;
    *min         = pop[1]->fitness;
    *max         = pop[1]->fitness;
    for (j=2; j<=popsize; j++)
    {
        (*sumfitness) += pop[j]->fitness;
        if ( (pop[j]->fitness) > (*max) )
            (*max) = pop[j]->fitness;
        if ( (pop[j]->fitness) < (*min) )
            (*min) = pop[j]->fitness;
    }
    (*avg) = (*sumfitness) / popsize;
}

// Initializations
void InitData(void)
{
    char ch;
    int j;
    clrscr();
    printf( "Genetic Algorithm \n");
    printf( "Enter population size = ");
    scanf( "%d", &popsize);
    printf( "Enter chromosome length = ");
    scanf( "%d", &lchrom);
    printf( "Enter max. generations = ");
    scanf( "%d", &maxgen);
    printf( "Enter crossover probability = ");
    scanf( "%lf", &pcross);
    printf( "Enter mutation probability = ");
    scanf( "%lf", &pmutation);
    clrscr();
    Randomize_Knuth();
    // Initiliaze Counters
    nmutation = 0;
    ncross    = 0;
    COEF_NORM = Power( 2, lchrom);
}

void InitReport(FILE *out)
{
    printf( " GA Parameters \n");
    printf( " Population Size           %d \n",      popsize);
    printf( " Maximum # of generation         %d \n",      lchrom);
    printf( " Crossover probability            %5.4f \n",    pcross);
    printf( " Mutation probability             %5.4f \n",    pmutation);
    printf( " \n Inital Generation Statistics \n");
    printf( " Initial population maximum fitness %5.2f \n", max);
    printf( " Initial population average fitness %5.2f \n", avg);
    printf( " Initial population minimum fitness %5.2f \n", min);
    printf( " Initial population sum of fitness %5.2f \n",
sumfitness);
    fprintf( out, " GA Parameters \n");
    fprintf( out, " Population Size           %d \n",      popsize);
    fprintf( out, " Maximum # of generation         %d \n",      lchrom);
    fprintf( out, " Crossover probability            %5.4f \n",    pcross);
    fprintf( out, " Mutation probability             %5.4f \n",    pmutation);
    fprintf( out, " \n Inital Generation Statistics \n");
    fprintf( out, " Initial population maximum fitness %5.2f \n",
max);
    fprintf( out, " Initial population average fitness %5.2f \n",
avg);
}

```



```

    fprintf (out, " Initial population minimum fitness %5.2f \n",
min);
    fprintf (out, " Initial population sum of fitness  %5.2f \n\n\n",
sumfitness);
}
// Initiliaze a population at random
void InitPop (void)
{
    int j, j1;
    for (j=1; j<=popsize; j++)
    {
        for (j1=1; j1<=lchrom; j1++)
            oldpop[j]->chrom[j1] = Flip(0.5);          // A fair
coin tosses
        oldpop[j]->x = Decode(oldpop[j]->chrom, lchrom);    //
Decode string
        oldpop[j]->fitness = Objfunc(oldpop[j]->x);    // Evaluate
initial fitness
        oldpop[j]->parent1 = 0;
        oldpop[j]->parent2 = 0;
        oldpop[j]->xsite = 0;
    }
}
// Initiliaze Coordinator
void Initiliaze (void)
{
    InitFile();
    InitData();
    InitPop();
    Statistic (popsize, &max, &avg, &min, &sumfitness, oldpop);
    InitReport(arquivo);
}

void WriteChrom ( FILE *out, chromosome chrom, int lchrom)
{
    int j;
    for (j=1;lchrom; j>=1; j--)
        if (chrom[j])
            fprintf ( out, "1");
        else
            fprintf ( out, "0");
}

void Report (int gen, FILE *out, FILE *out2)
{
    const linelenght = 132;
    int j;
    fprintf ( out, "                                Population Report  \n
");
    fprintf ( out, "  Generation  %d
Generation  %d  \n ", gen-1, gen );
    fprintf ( out, " #           String                                x
fitness  #  parents xsite                                string
x      fitness  \n" );
    fprintf ( out, " -----
-----\n" );
    for (j=1; j<=popsize; j++)
    {
        fprintf ( out, "  %02d) ", j);

        WriteChrom ( out, oldpop[j]->chrom, lchrom);
        fprintf ( out, "  %06.04E  %06.04f  |", oldpop[j]->x, oldpop[j]-
>fitness);
        fprintf ( out, "  %02d) (%02d,%02d) %02d      ", j, newpop[j]-
>parent1, newpop[j]->parent2, newpop[j]->xsite);
        WriteChrom ( out, newpop[j]->chrom, lchrom);
        fprintf ( out, "  %06.04E      %06.04f  \n", newpop[j]->x,
newpop[j]->fitness);
    }

    fprintf ( out, " -----
-----\n " );
}

```

```

    fprintf ( out, " Note: Generation %02d Accumulated Statistics :
max = %06.04f min = %06.04f avg = %06.04f sum = %06.04f
nmutation = %03d ncross = %03d \n", gen, max, min, avg, sumfitness,
nmutation, ncross);
    fprintf ( out, " -----
----- \n\n%f" );
    fprintf ( out2, " %06.04f %06.04f %06.04f \n", max, min, avg
);
}
// Select a single individual via roulette wheel selection
int Select( int popsize, double sumfitness, population *pop)
{
    double randnum, partsum; // random point wheel selection
    int j;
    partsum = 0.0;
    j = 0;
    randnum = Random_Knuth() * sumfitness;
    do
    {
        j++;
        partsum += pop[j]->fitness;
    }
    while ((partsum < randnum) && (j < popsize));
    return (j);
}
allele Mutation ( allele alleleval, double pmutation, int *nmutation
)
{
    boolean mutate;
    mutate = Flip(pmutation);
    if (mutate)
    {
        (*nmutation)++;
        return( !alleleval );
    }
    else
        return ( alleleval );
}
void Crossover ( chromosome parent1, chromosome parent2,
                chromosome child1, chromosome child2,
                int *lchrom, int *ncross, int *nmutation, int
*jcross,
                double *pcross, double *pmutation )
{
    int j;
    if ( Flip(*pcross) )
    {
        *jcross = Rnd(1, (*lchrom)-1);
        *ncross = (*ncross) + 1;
    }
    else
        *jcross = *lchrom;
    // 1st exchange, 1 to 1 and 2 to 2
    for(j=1; j<=(*jcross); j++)
    {
        *(child1+j) = Mutation( parent1[j], *pmutation, nmutation );
        *(child2+j) = Mutation( parent2[j], *pmutation, nmutation );
    }
    // 2nd exchange 1 to 2 and 2 to 1
    if ( (*jcross) != (*lchrom) )
    {
        for (j=(*jcross)+1; j<=(*lchrom); j++)
        {
            *(child1+j) = Mutation( parent2[j], *pmutation, nmutation );
            *(child2+j) = Mutation( parent1[j], *pmutation, nmutation );
        }
    }
}
void Crossover2 ( chromosome parent1, chromosome parent2,
                chromosome child1, chromosome child2,
                int *lchrom, int *ncross, int *nmutation, int
*jcross,
                double *pcross, double *pmutation )
{

```

```

int j;
chromosome tmp;
// exchange
for(j=1; j<=(*lchrom); j++)
{
    if ( Flip(*pcross) )
    {
        *tmp = parent1[j];
        *(child1+j) = parent2[j];
        *(child2+j) = *tmp;
    }
    Mutation( parent1[j], *pmutation, nmutation );
    Mutation( parent2[j], *pmutation, nmutation );
}
}
// Create a new generation through select, crossover and mutation
void Generation (void)
{
    int j, matel, mate2, jcross;
    j = 1;
    do
    {
        matel = Select (popsize, sumfitness, oldpop);
        mate2 = Select (popsize, sumfitness, oldpop);
        Crossover2( oldpop[matel]->chrom, oldpop[mate2]->chrom,
                    newpop[j]->chrom, newpop[j+1]->chrom,
                    &lchrom, &ncross, &nmutation, &jcross, &pcross,
                    &pmutation );
        newpop[j]->x = Decode( newpop[j]->chrom, lchrom );
        newpop[j]->fitness = Objfunc( newpop[j]->x );
        newpop[j]->parent1 = matel;
        newpop[j]->parent2 = mate2;
        newpop[j]->xsite = jcross;
        newpop[j+1]->x = Decode( newpop[j+1]->chrom, lchrom );
        newpop[j+1]->fitness = Objfunc( newpop[j+1]->x );
        newpop[j+1]->parent1 = matel;
        newpop[j+1]->parent2 = mate2;
        newpop[j+1]->xsite = jcross;
        j = j + 2;
    } while (j<=popsize);
}
void Pause ( int pauselength )
{
    const MAXPAUSE = 2500;
    int j, j1;
    double x;
    for (j=1; j<= pauselength; j++)
        for (j1=1; j1<=MAXPAUSE; j1++)
            x = 0.0 + 1.0;
}
// Flip a biased coin - true if heads
boolean Flip ( double probability )
{
    if ( probability == 1.0 )
        return ( TRUE );
    else
        return ( Random_Knuth() <= probability );
}
// Pick a random integer between low and high
int Rnd ( int low, int high )
{
    int i;
    if ( low >= high )
        i = low;
    else
    {
        i = int ( Random_Knuth() * (high - low - 1) + low );
        if ( i > high )
            i = high;
    }
    return (i);
}
double Power( double base, double expo)
{
    return ( exp( expo*log(base) ) );
}

```

```

}
void InitFile( void )
{
    // Initialize data file
    /* open a file for update */
    arquivo = fopen( "saida.txt", "w");
    if (ferror(arquivo)) /* test for an error on the stream */
    {
        /* display an error message */
        printf("Error reading from %s \n", arquivo);
        /* reset the error and EOF indicators */
        clearerr(arquivo);
    }
    arquivo2 = fopen( "resumo.txt", "w");
    if (ferror(arquivo2)) /* test for an error on the stream */
    {
        /* display an error message */
        printf("Error reading from %s \n", arquivo2);
        /* reset the error and EOF indicators */
        clearerr(arquivo2);
    }
    fprintf ( arquivo2, "                                Population Report
\n ");
    fprintf ( arquivo2, "      Max      Min      Avg      \n" );
}
// Calculate scalling coefficients for linear scalling
void PreScale ( double umax, double uavg, double umin, double *a,
double *b )
{
    const float fmultiple = 2.0;
    double delta;
    if ( umin > (fmultiple * uavg - umax)/(fmultiple - 1.0) ) // Non-
negative test
    {
        // Normal Scalling
        delta = umax - uavg;
        *a = (fmultiple - 1.0) * uavg / delta;
        *b = uavg * (umax - fmultiple * uavg) / delta;
    }
    else
    {
        // Scale as much as possible
        delta = uavg - umin;
        *a = uavg / delta;
        *b = -umin * uavg / delta;
    }
}
// Scale an objective function
double Scale ( double u, double a, double b )
{
    return ( a * u + b);
}
void ScalePop ( int popsize, double *max, double *avg, double *min,
double * sumfitness, population *pop)
{
}

```

Glossário dos Principais Conceitos de Genética Aplicados aos GAs

A genética é uma ciência que estuda a transferência de informações biológicas de célula para célula, de pais para filhos e, assim, de geração para geração, tratando dos mecanismos inerentes que controlam a constância e mudanças dos seres vivos. Esta ciência nasceu com a descoberta dos princípios mendelianos no início do século. A nível de engenharia, a analogia entre os sistemas naturais de adaptação e os sistemas artificiais conduziram ao desenvolvimento dos GAs.

Abaixo são definidos alguns conceitos utilizados pela genética e pelos GAs:

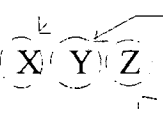
- a) adaptabilidade: o número da descendência deixada por um indivíduo quando comparada com a média da população ou comparado com indivíduos de genótipos diferentes. valor reprodutivo determinado por um genótipo em uma população;
- b) adaptação: ajustamento de um organismo ou população ao ambiente;
- c) alelo: membro de um par, ou series de genes alternativos que ocorrem em um dado *locus* (vide definição na alínea *r*) em um cromossomo; uma forma contrastante de genes;
- d) ambiente: o agregado de todas as condições externas e influências que afetam a vida e o desenvolvimento de um organismo;

- e) atavismo: reaparecimento de um caráter ancestral depois de várias gerações. Um gene que vem se expressar depois de um período de não expressão por causa da excessividade, ou de outro efeito que o mascare;
- f) biótipo: raça ou linhagem fisiologicamente distinta dentro de uma espécie morfológica. Uma população de indivíduos com constituição genética idêntica;
- g) caráter: um dos muitos detalhes de estrutura, forma ou substância que forma um organismo individual;
- h) carga genética: a proporção pela qual a adaptabilidade do genótipo ótimo é diminuída por genes deletérios; expressa-se em equivalentes letais ou “mortes gênicas”;
- i) conjunto gênico: soma total de todos os alelos diferentes nos membros de uma população que se cruzam em um determinado tempo;
- j) cromossomos: estruturas nucleoprotéicas que carregam os genes;
- k) *crossing over*: processo inferido geneticamente pela associação recente de genes ligados e demonstrados citologicamente a partir de associações recentes de pares de cromossomos. Resulta em uma troca de genes, e portanto produz combinações que diferem das características dos progenitores;
- l) expressividade: grau de expressão de um caráter controlado por um gene;
- m) fenótipo: características de um indivíduo observáveis, ou discerníveis por outros métodos;

- n) gene: um determinador hereditário que especifica uma função biológica; uma unidade de herança (DNA) localizada em um lugar fixo do cromossomo;
- o) genética: a ciência da hereditariedade e variação;
- p) genoma: um grupo completo de cromossomos herdado como uma unidade a partir de um progenitor;
- q) genótipo: a constituição genética de um organismo;
- r) locus: Uma posição fixa em um cromossomo que é ocupada por um determinado gene ou um dos seus alelos;
- s) mutação: Uma alteração em um locus particular do DNA de um organismo.
- t) seleção: reprodução diferencial de genótipos diferentes. O fator mais importante que altera as frequências dos alelos e genótipos em uma grande população, influenciando assim mudanças evolucionárias;

ANEXO IV

Tempos e Máquinas

Obs: _____ a) referência c) tempo de operação
 _____ b) código da máquina (dado em horas)

 • 998 - fim de seqüência
 • 999 - fim do arquivo

1 37 2	29 47 2	57 3 0
2 30 12.5	30 3 0	58 998 0
3 31 21	31 998 0	59 39 0.3
4 37 3	32 37 0.2	60 3 0
5 3 0	33 3 0	61 998 0
6 998 0	34 998 0	62 49 1.5
7 37 7.5	35 3 12	63 47 0.5
8 39 3	36 61 0	64 3 0
9 61 3	37 998 0	65 998 0
10 3 0	38 3 0.8	66 40 0.97
11 998 0	39 61 0	67 3 0
12 3 17	40 998 0	68 998 0
13 61 6	41 61 4.6	69 3 23
14 3 0	42 3 0	70 99 60
15 998 0	43 998 0	71 99 0
16 34 0.5	44 37 1.2	72 998 0
17 27 2.8	45 40 0.7	73 49 1.5
18 39 6	46 47 1	74 99 0
19 3 0	47 3 0	75 998 0
20 998 0	48 998 0	76 99 60
21 33 0.3	49 3 75	77 99 55
22 32 1	50 3 10	78 99 0
23 3 0	51 3 0	79 998 0
24 998 0	52 998 0	80 33 1.5
25 3 7	53 34 2.3	81 99 0
26 3 0	54 3 0	82 998 0
27 998 0	55 998 0	83 99 50
28 49 2.8	56 3 28	84 99 10

ANEXO IV

85 99 0	121 61 3	157 40 1
86 998 0	122 3 0	158 47 1
87 49 2	123 998 0	159 3 0
88 47 0.5	124 3 20	160 998 0
89 4 0	125 61 7	161 3 85
90 998 0	126 3 0	162 3 15
91 39 3.5	127 998 0	163 3 0
92 4 0	128 34 0.7	164 998 0
93 998 0	129 27 2.8	165 34 3.5
94 37 0.5	130 39 6	166 3 0
95 4 0	131 3 0	167 998 0
96 998 0	132 998 0	168 3 24
97 4 15	133 33 0.4	169 3 0
98 99 0	134 32 1	170 998 0
99 998 0	135 3 0	171 39 0.5
100 49 4.5	136 998 0	172 3 0
101 47 1.3	137 3 10	173 998 0
102 99 0	138 3 0	174 49 1.5
103 998 0	139 998 0	175 47 1.5
104 27 4	140 49 3	176 3 0
105 99 0	141 47 2	177 998 0
106 998 0	142 3 0	178 40 1
107 99 20	143 998 0	179 3 0
108 99 25	144 37 0.3	180 998 0
109 99 0	145 3 0	181 3 27
110 998 0	146 998 0	182 99 63
111 99 45	147 3 15	183 99 0
112 999 0	148 61 0	184 998 0
113 37 2.5	149 998 0	185 49 1.5
114 30 15	150 3 1	186 99 0
115 31 22	151 61 0	187 998 0
116 37 3.5	152 998 0	188 99 104
117 3 0	153 61 5	189 99 82
118 998 0	154 3 0	190 99 0
119 37 10	155 998 0	191 998 0
120 39 3	156 37 1.5	192 33 1.5

ANEXO IV

193 99 0	229 3 0	265 61 5.5
194 998 0	230 998 0	266 3 0
195 99 65	231 37 12.8	267 998 0
196 99 12.5	232 39 4	268 37 1
197 99 0	233 61 4	269 40 1.5
198 998 0	234 3 0	270 47 1
199 49 2	235 998 0	271 3 0
200 47 0.5	236 3 24	272 998 0
201 4 0	237 61 9	273 3 100
202 998 0	238 3 0	274 3 18
203 39 3.5	239 998 0	275 3 0
204 4 0	240 34 0.75	276 998 0
205 998 0	241 27 2.8	277 34 4
206 37 0.5	242 39 6.5	278 3 0
207 4 0	243 3 0	279 998 0
208 998 0	244 998 0	280 3 31
209 4 15	245 33 0.5	281 3 0
210 99 0	246 32 1.5	282 998 0
211 998 0	247 3 0	283 39 0.5
212 49 5.5	248 998 0	284 3 0
213 47 1.4	249 3 13	285 998 0
214 99 0	250 3 0	286 49 1.8
215 998 0	251 998 0	287 47 1.6
216 27 5	252 49 4	288 3 0
217 99 0	253 47 2	289 998 0
218 998 0	254 3 0	290 40 1.3
219 99 18.5	255 998 0	291 3 0
220 99 19.5	256 37 0.35	292 998 0
221 99 0	257 3 0	293 3 31
222 998 0	258 998 0	294 99 75
223 99 60	259 3 21	295 99 0
224 999 0	260 61 0	296 998 0
225 37 4	261 998 0	297 49 1.7
226 30 17	262 3 1	298 99 0
227 31 27	263 61 0	299 998 0
228 37 3.5	264 998 0	300 99 108

ANEXO IV

301 99 80	337 37 7	373 61 0
302 99 0	338 30 20	374 998 0
303 998 0	339 31 40	375 3 1.5
304 33 1.8	340 37 4.5	376 61 0
305 99 0	341 3 0	377 998 0
306 998 0	342 998 0	378 61 6
307 99 70	343 37 23	379 3 0
308 99 14	344 61 2	380 998 0
309 99 0	345 39 5	381 37 3
310 998 0	346 61 6	382 40 1.5
311 49 2	347 3 0	383 47 2
312 47 0.5	348 998 0	384 3 0
313 4 0	349 3 30	385 998 0
314 998 0	350 61 11	386 3 150
315 39 3.5	351 3 0	387 3 20
316 4 0	352 998 0	388 3 0
317 998 0	353 34 0.8	389 998 0
318 37 0.5	354 27 4.6	390 34 6.7
319 4 0	355 39 7	391 3 0
320 998 0	356 3 0	392 998 0
321 4 15	357 998 0	393 3 50
322 99 0	358 33 0.6	394 3 0
323 998 0	359 32 1.5	395 998 0
324 49 5.5	360 3 0	396 39 0.7
325 47 1.8	361 998 0	397 3 0
326 99 0	362 3 15	398 998 0
327 998 0	363 3 0	399 49 2
328 27 6	364 998 0	400 47 2.5
329 99 0	365 49 4	401 3 0
330 998 0	366 47 3	402 998 0
331 99 20	367 3 0	403 40 1
332 99 26	368 998 0	404 3 0
333 99 0	369 37 0.4	405 998 0
334 998 0	370 3 0	406 3 35
335 99 75	371 998 0	407 99 93
336 999 0	372 3 26	408 99 0

ANEXO IV

409 998 0	445 99 30	481 37 0.5
410 49 2	446 99 0	482 3 0
411 99 0	447 998 0	483 998 0
412 998 0	448 99 85	484 3 27
413 99 110	449 999 0	485 61 0
414 99 90	450 37 10	486 998 0
415 99 0	451 30 25	487 3 1.8
416 998 0	452 31 45	488 61 0
417 33 1.8	453 37 5	489 998 0
418 99 0	454 3 0	490 61 8
419 998 0	455 998 0	491 3 0
420 99 82	456 37 25	492 998 0
421 99 15	457 39 6	493 37 2.5
422 99 0	458 61 8	494 40 1.5
423 998 0	459 3 0	495 47 3
424 49 3	460 998 0	496 3 0
425 47 0.9	461 3 32	497 998 0
426 4 0	462 61 15	498 3 160
427 998 0	463 3 0	499 3 23
428 39 4.5	464 998 0	500 3 0
429 4 0	465 34 0.6	501 998 0
430 998 0	466 27 5	502 34 7.1
431 37 1	467 39 9	503 3 0
432 4 0	468 3 0	504 998 0
433 998 0	469 998 0	505 3 62
434 4 17	470 33 0.8	506 3 0
435 99 0	471 32 3	507 998 0
436 998 0	472 3 0	508 39 0.8
437 49 5.5	473 998 0	509 3 0
438 47 1.8	474 3 22	510 998 0
439 99 0	475 3 0	511 49 2.5
440 998 0	476 998 0	512 47 2.6
441 27 7	477 49 4.5	513 3 0
442 99 0	478 47 4	514 998 0
443 998 0	479 3 0	515 40 1.78
444 99 25	480 998 0	516 3 0

ANEXO IV

517 998 0	553 27 8	589 49 5.5
518 3 42	554 99 0	590 47 5
519 99 100	555 998 0	591 3 0
520 99 0	556 99 30	592 998 0
521 998 0	557 99 35	593 37 0.6
522 49 3	558 99 0	594 3 0
523 99 0	559 998 0	595 998 0
524 998 0	560 99 100	596 3 35
525 99 128	561 999 0	597 61 0
526 99 102	562 37 12	598 998 0
527 99 0	563 30 27	599 99
528 998 0	564 31 70	
529 33 2.4	565 37 7	
530 99 0	566 3 0	
531 998 0	567 998 0	
532 99 90	568 37 30	
533 99 20	569 39 7	
534 99 0	570 61 9	
535 998 0	571 3 0	
536 49 3	572 998 0	
537 47 0.9	573 3 50	
538 4 0	574 61 18	
539 998 0	575 3 0	
540 39 4.5	576 998 0	
541 4 0	577 34 0.8	
542 998 0	578 27 5.3	
543 37 1	579 39 12.34	
544 4 0	580 3 0	
545 998 0	581 998 0	
546 4 17	582 33 1	
547 99 0	583 32 4	
548 998 0	584 3 0	
549 49 6	585 998 0	
550 47 2	586 3 27	
551 99 0	587 3 0	
552 998 0	588 998 0	

Dependências

Obs:

X

YYYYYYYYYYY

a) referência

b) componentes

(código das peças)

1 0 0 0 0 0 0 0 0 0 0	30 28 29 0 0 0 0 0 0 0 0	60 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0	31 23 27 30 0 0 0 0 0 0 0	61 0 0 0 0 0 0 0 0 0 0
3 1 2 0 0 0 0 0 0 0 0	33 0 0 0 0 0 0 0 0 0 0	62 28 29 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0	34 0 0 0 0 0 0 0 0 0 0	63 23 27 30 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0	35 1 2 0 0 0 0 0 0 0 0	65 0 0 0 0 0 0 0 0 0 0
6 4 5 0 0 0 0 0 0 0 0	36 0 0 0 0 0 0 0 0 0 0	66 0 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0	37 0 0 0 0 0 0 0 0 0 0	67 1 2 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0	38 4 5 0 0 0 0 0 0 0 0	68 0 0 0 0 0 0 0 0 0 0
9 7 8 0 0 0 0 0 0 0 0	39 0 0 0 0 0 0 0 0 0 0	69 0 0 0 0 0 0 0 0 0 0
10 0 0 0 0 0 0 0 0 0 0	40 0 0 0 0 0 0 0 0 0 0	70 4 5 0 0 0 0 0 0 0 0
11 9 10 0 0 0 0 0 0 0 0	41 7 8 0 0 0 0 0 0 0 0	71 0 0 0 0 0 0 0 0 0 0
12 0 0 0 0 0 0 0 0 0 0	42 0 0 0 0 0 0 0 0 0 0	72 0 0 0 0 0 0 0 0 0 0
13 3 6 11 12 0 0 0 0 0 0	43 9 10 0 0 0 0 0 0 0 0	73 7 8 0 0 0 0 0 0 0 0
14 0 0 0 0 0 0 0 0 0 0	44 0 0 0 0 0 0 0 0 0 0	74 0 0 0 0 0 0 0 0 0 0
15 13 14 0 0 0 0 0 0 0 0	45 3 6 11 12 0 0 0 0 0 0	75 9 10 0 0 0 0 0 0 0 0
16 0 0 0 0 0 0 0 0 0 0	46 0 0 0 0 0 0 0 0 0 0	76 0 0 0 0 0 0 0 0 0 0
17 0 0 0 0 0 0 0 0 0 0	47 13 14 0 0 0 0 0 0 0 0	77 3 6 11 12 0 0 0 0 0 0
18 0 0 0 0 0 0 0 0 0 0	48 0 0 0 0 0 0 0 0 0 0	78 0 0 0 0 0 0 0 0 0 0
19 15 16 17 18 0 0 0 0 0 0	49 0 0 0 0 0 0 0 0 0 0	79 13 14 0 0 0 0 0 0 0 0
20 0 0 0 0 0 0 0 0 0 0	50 0 0 0 0 0 0 0 0 0 0	80 0 0 0 0 0 0 0 0 0 0
21 19 20 0 0 0 0 0 0 0 0	51 15 16 17 18 0 0 0 0 0 0	81 0 0 0 0 0 0 0 0 0 0
22 0 0 0 0 0 0 0 0 0 0	52 0 0 0 0 0 0 0 0 0 0	82 0 0 0 0 0 0 0 0 0 0
23 21 22 0 0 0 0 0 0 0 0	53 19 20 0 0 0 0 0 0 0 0	83 15 16 17 18 0 0 0 0 0 0
24 0 0 0 0 0 0 0 0 0 0	54 0 0 0 0 0 0 0 0 0 0	84 0 0 0 0 0 0 0 0 0 0
25 0 0 0 0 0 0 0 0 0 0	55 21 22 0 0 0 0 0 0 0 0	85 19 20 0 0 0 0 0 0 0 0
26 0 0 0 0 0 0 0 0 0 0	56 0 0 0 0 0 0 0 0 0 0	86 0 0 0 0 0 0 0 0 0 0
27 24 25 26 0 0 0 0 0 0 0	57 0 0 0 0 0 0 0 0 0 0	87 21 22 0 0 0 0 0 0 0 0
28 0 0 0 0 0 0 0 0 0 0	58 0 0 0 0 0 0 0 0 0 0	88 0 0 0 0 0 0 0 0 0 0
29 0 0 0 0 0 0 0 0 0 0	59 24 25 26 0 0 0 0 0 0 0	89 0 0 0 0 0 0 0 0 0 0

90 0000000000	125 0000000000	161 0000000000
91 24 25 26 0000000	126 28 29 00000000	162 0000000000
92 0000000000	127 23 27 30 00000000	163 1 2 00000000
93 0000000000	129 0000000000	164 0000000000
94 28 29 00000000	130 0000000000	165 0000000000
95 23 27 30 0000000	131 1 2 00000000	166 4 5 00000000
97 0000000000	132 0000000000	167 0000000000
98 0000000000	133 0000000000	168 0000000000
99 1 2 00000000	134 4 5 00000000	169 7 8 00000000
100 0000000000	135 0000000000	170 0000000000
101 0000000000	136 0000000000	171 9 10 00000000
102 4 5 00000000	137 7 8 00000000	172 0000000000
103 0000000000	138 0000000000	173 3 6 11 12 000000
104 0000000000	139 9 10 00000000	174 0000000000
105 7 8 00000000	140 0000000000	175 13 14 00000000
106 0000000000	141 3 6 11 12 000000	176 0000000000
107 9 10 00000000	142 0000000000	177 0000000000
108 0000000000	143 13 14 00000000	178 0000000000
109 3 6 11 12 000000	144 0000000000	179 15 16 17 18 00000
110 0000000000	145 0000000000	0
111 13 14 00000000	146 0000000000	180 0000000000
112 0000000000	147 15 16 17 18 00000	181 19 20 00000000
113 0000000000	0	182 0000000000
114 0000000000	148 0000000000	183 21 22 00000000
115 15 16 17 18 00000	149 19 20 00000000	184 0000000000
0	150 0000000000	185 0000000000
116 0000000000	151 21 22 00000000	186 0000000000
117 19 20 00000000	152 0000000000	187 24 25 26 0000000
118 0000000000	153 0000000000	188 0000000000
119 21 22 00000000	154 0000000000	189 0000000000
120 0000000000	155 24 25 26 0000000	190 28 29 00000000
121 0000000000	156 0000000000	191 23 27 30 0000000
122 0000000000	157 0000000000	193 0000000000
123 24 25 26 0000000	158 28 29 00000000	194 0000000000
124 0000000000	159 23 27 30 0000000	195 1 2 00000000

196 0 0 0 0 0 0 0 0 0 0	231 0 0 0 0 0 0 0 0 0 0	266 0 0 0 0 0 0 0 0 0 0
197 0 0 0 0 0 0 0 0 0 0	232 0 0 0 0 0 0 0 0 0 0	267 9 10 0 0 0 0 0 0 0 0
198 4 5 0 0 0 0 0 0 0 0	233 7 8 0 0 0 0 0 0 0 0	268 0 0 0 0 0 0 0 0 0 0
199 0 0 0 0 0 0 0 0 0 0	234 0 0 0 0 0 0 0 0 0 0	269 3 6 11 12 0 0 0 0 0 0
200 0 0 0 0 0 0 0 0 0 0	235 9 10 0 0 0 0 0 0 0 0	270 0 0 0 0 0 0 0 0 0 0
201 7 8 0 0 0 0 0 0 0 0	236 0 0 0 0 0 0 0 0 0 0	271 13 14 0 0 0 0 0 0 0 0
202 0 0 0 0 0 0 0 0 0 0	237 3 6 11 12 0 0 0 0 0 0	272 0 0 0 0 0 0 0 0 0 0
203 9 10 0 0 0 0 0 0 0 0	238 0 0 0 0 0 0 0 0 0 0	273 0 0 0 0 0 0 0 0 0 0
204 0 0 0 0 0 0 0 0 0 0	239 13 14 0 0 0 0 0 0 0 0	274 0 0 0 0 0 0 0 0 0 0
205 3 6 11 12 0 0 0 0 0 0	240 0 0 0 0 0 0 0 0 0 0	275 15 16 17 18 0 0 0 0 0 0
206 0 0 0 0 0 0 0 0 0 0	241 0 0 0 0 0 0 0 0 0 0	0
207 13 14 0 0 0 0 0 0 0 0	242 0 0 0 0 0 0 0 0 0 0	276 0 0 0 0 0 0 0 0 0 0
208 0 0 0 0 0 0 0 0 0 0	243 15 16 17 18 0 0 0 0 0 0	277 19 20 0 0 0 0 0 0 0 0
209 0 0 0 0 0 0 0 0 0 0	0	278 0 0 0 0 0 0 0 0 0 0
210 0 0 0 0 0 0 0 0 0 0	244 0 0 0 0 0 0 0 0 0 0	279 21 22 0 0 0 0 0 0 0 0
211 15 16 17 18 0 0 0 0 0 0	245 19 20 0 0 0 0 0 0 0 0	280 0 0 0 0 0 0 0 0 0 0
0	246 0 0 0 0 0 0 0 0 0 0	281 0 0 0 0 0 0 0 0 0 0
212 0 0 0 0 0 0 0 0 0 0	247 21 22 0 0 0 0 0 0 0 0	282 0 0 0 0 0 0 0 0 0 0
213 19 20 0 0 0 0 0 0 0 0	248 0 0 0 0 0 0 0 0 0 0	283 24 25 26 0 0 0 0 0 0 0 0
214 0 0 0 0 0 0 0 0 0 0	249 0 0 0 0 0 0 0 0 0 0	284 0 0 0 0 0 0 0 0 0 0
215 21 22 0 0 0 0 0 0 0 0	250 0 0 0 0 0 0 0 0 0 0	285 0 0 0 0 0 0 0 0 0 0
216 0 0 0 0 0 0 0 0 0 0	251 24 25 26 0 0 0 0 0 0 0 0	286 28 29 0 0 0 0 0 0 0 0
217 0 0 0 0 0 0 0 0 0 0	252 0 0 0 0 0 0 0 0 0 0	287 23 27 30 0 0 0 0 0 0 0 0
218 0 0 0 0 0 0 0 0 0 0	253 0 0 0 0 0 0 0 0 0 0	289 0 0 0 0 0 0 0 0 0 0
219 24 25 26 0 0 0 0 0 0 0 0	254 28 29 0 0 0 0 0 0 0 0	290 0 0 0 0 0 0 0 0 0 0
220 0 0 0 0 0 0 0 0 0 0	255 23 27 30 0 0 0 0 0 0 0 0	291 1 2 0 0 0 0 0 0 0 0
221 0 0 0 0 0 0 0 0 0 0	257 0 0 0 0 0 0 0 0 0 0	292 0 0 0 0 0 0 0 0 0 0
222 28 29 0 0 0 0 0 0 0 0	258 0 0 0 0 0 0 0 0 0 0	293 3 4 0 0 0 0 0 0 0 0
223 23 27 30 0 0 0 0 0 0 0 0	259 1 2 0 0 0 0 0 0 0 0	294 0 0 0 0 0 0 0 0 0 0
225 0 0 0 0 0 0 0 0 0 0	260 0 0 0 0 0 0 0 0 0 0	295 5 6 0 0 0 0 0 0 0 0
226 0 0 0 0 0 0 0 0 0 0	261 0 0 0 0 0 0 0 0 0 0	296 0 0 0 0 0 0 0 0 0 0
227 1 2 0 0 0 0 0 0 0 0	262 4 5 0 0 0 0 0 0 0 0	297 0 0 0 0 0 0 0 0 0 0
228 0 0 0 0 0 0 0 0 0 0	263 0 0 0 0 0 0 0 0 0 0	298 7 8 9 0 0 0 0 0 0 0 0
229 0 0 0 0 0 0 0 0 0 0	264 0 0 0 0 0 0 0 0 0 0	299 0 0 0 0 0 0 0 0 0 0
230 4 5 0 0 0 0 0 0 0 0	265 7 8 0 0 0 0 0 0 0 0	300 10 11 0 0 0 0 0 0 0 0

301 0 0 0 0 0 0 0 0 0 0

302 12 13 0 0 0 0 0 0 0 0

303 0 0 0 0 0 0 0 0 0 0

REFERÊNCIAS BIBLIOGRÁFICAS

- ANSOFF, I., Estratégia Empresarial, Ed. McGraw-Hill do Brasil, São Paulo, 1977.
- ARBEL, A. e SEIDMANN, A., Performance Evaluation of FMS. IEEE Transactions on Systems, Man and Cybernetics, July-Aug., 1984, pp. 606-617.
- ARIZONO, I. et al., Scheduling for Minimizing Total actual Flow Time by Neural Networks, International Journal of Production Research, V.30, 3, pg. 503-511, 1992.
- AROSIO, M., SIANESI, A., A Heuristic Algorithm for Master Production schedule Generation with Finite capacity and Sequence Dependent Setups, International journal of Production Research, V.31, 3, pg. 531-553, 1993.
- BAGLEY, J. D., The Behaviour of Adaptive Systems Which Employ Genetic and Correlation Algorithms, Doctoral Dissertation, University of Michigan, USA, 1967.
- BARRICELLI, N. A., Symbiogenetic Evolution processes Realized by Artificial Methods, Methodos, 9, pg. 134-182, 1957.
- BARRICELLI, N. A., Numerical testing of Evolution Theories, ACTA Biotheoretica, pg. 69-126, 1962.
- BEASLEY, D. et al., An Overview of Genetic Algorithms, University Computing, 15, pg. 58-69, 1993.
- BILBERG, A., ALTING, L., When Simulation Takes Control, Journal of Manufacturing Systems, V.10, 4, pg. 179-192, 1992.
- BLACKSTONE, J.H. et al., A State-of-Art Survey of Dispatching Rules for Manufacturing Job Shop Operations, International Journal of Production Research, 20, pg. 27-45, 1982.

REFERÊNCIAS BIBLIOGRÁFICAS

- BOZER, Y.A., Guided Vehicle Systems: Information/Control System Implications of Alternative Design and Operation Strategies", In: Advanced Information Technologies for Industrial Material Flow Systems, Ed. S.Y. Nof and C.L. Moodie, 1989.
- CARRIE, A.S. e PETSOPOULOS, A.C., Operation Sequencing in a FMS, Robotica, 3, 259-264, 1985.
- CAVICCHIO, D. J., Adaptive Search Using Simulated Evolution, Doctoral Dissertation, University of Michigan, USA, 1970.
- CHO, H., WYSK, R.A., A Robust Adaptive for an Intelligent Workstation Controller, International Journal of Production Research, V. 31, 4, pg. 771-789, 1993.
- CO, H.C., JAW, T.J., Sequencing in Manufacturing Systems and Other Short Queue Length Systems, Journal of Manufacturing Systems, 7, 1-9, 1988
- DAR-EL, E.M., WYSK, R.A., A Job-Shop Scheduling - A Systematic Approach, Journal of Manufacturing Systems, 1, pg.77-88, 1982.
- DAVIS, L., Job Shop Scheduling with Genetic algorithms, In: Proceedings of the First International Conference on Genetic Algorithms, pg. 136-140, 1985.
- EGBELU, P.J., e TANCHOCO, J.M.A., Characterization of the Automatic Guided Vehicle Dispatching Rules, International Journal of Production Research, 22, 359-374, 1984.
- FRASER, A. S., Simulation of Genetic systems by Automatic Digital Computers, Biometrical Genetics, pg. 70-83, 1960.
- FRASER, A. S., Simulation of Genetic Systems, Journal of Theoretical Biology, 2, pg. 329-346, 1962.
- FREITAS, P. J., Um Sistema Inteligente de Simulação para Avaliação de Desempenho Operacional de Um Sistema Flexível de Manufatura, Tese de Doutorado, Departamento de Engenharia de Produção e Sistemas, UFSC, Florianópolis, 1994.

REFERÊNCIAS BIBLIOGRÁFICAS

- GAULTHER, F., Tese de Doutorado, Departamento de Engenharia de Produção e Sistemas, UFSC, Florianópolis, 1994.
- GOLDBERG, D. E., Computer-Aided Pipeline Operation Using Genetic Algorithms and Rule Learning, Doctoral Dissertation, University of Michigan, USA, 1983.
- GOLDBERG, D. E., Genetic algorithms in Search, Optimization, and Machine Learning, Ed. Addison-Wesley Publishing Company, Inc., 1989.
- GUPTA, Y. P. et al., A review of Scheduling Rules in Flexible Manufacturing Systems, International Journal of Computer Integrated Manufacturing, 2, pg. 356-377, 1990.
- GUPTA, Y.P. e GOYAL, S., Flexibility Trade-Offs in a Random FMS: A Simulation Study. International Journal of Production Research. V-30, Nº 3, March, 527-557, 1992.
- GUPTA, Y. P. et al., Operational Planning and Scheduling Problems in Advanced Manufacturing Systems, International Journal of Production Research, V. 31, 4, pg. 869-900, 1993.
- GUPTA, Y.P. et al., Multi-Criterion Approaches to FMS Scheduling Problem, International Journal of Production Economics, 22, 13-31, 1991.
- HOLLAND, J. H., Outline for a Logical Theory of Adaptive Systems, Journal of the Association for Computer Machinery, 3, pg. 297-314, 1962.
- HOLLSTIEN, R. B., Artificial Genetic Adaptation in Computer Control Systems, Doctoral Dissertation, University of Michigan, USA, 1971.
- HOPFIELD, J. J., TANK, D. W., Neural Computation of Decisions in Optimization Problems, Biological Cybernetics, 52, pg. 141-152, 1985.
- HUSBAND et al., Genetic Algorithms, Production Plan Optimization and Scheduling, Proceedings of Third International Conference on Genetic Algorithms, 1989, San maleo, pp. 160-169.

REFERÊNCIAS BIBLIOGRÁFICAS

- KOSKO, B., Neural Networks and Fuzzy Systems, Ed. Prentice-Hall, Inc., 1992
- KUSIAK, A., Flexible Manufacturing Systems: A Structural Approach, International Journal of Production Research, 23, 1057-1073, 1985.
- LECOMTE, C., DEJAX, P., A Decision Support System for the Production Control of Make-to-Order Workshop, In Proceedings of First IFOPS Specialized Conference on Decision Support Systems, March 26-29, Bruges, Belgium, 1991.
- LEUNG, L. C., MAHESHWARI, S. K. e MILLER, W. A., Concurrent Part Routing and Tool Allocation in FMS with Material Handling Considerations, Working Paper, Department of Industrial and Management Systems Engineering, U.S.F., Tampa, USA, 1990.
- MAHESHWARI, S. K. , Planning and Operational Control of FMS with Material Handling Considerations, Ph.D. Dissertation, Department of Industrial and Management Systems Engineering, U.S.F., Tampa, USA, 1992.
- MARTIN, F. G., COCKERHAM, C. C., High Speed Selection Studies", Biometrical Genetics, pg. 35-45, 1960.
- MINGA, A. K., Genetic Algorithms in Aerospace Design, 1986.
- MONTAZERI, M. e VAN WASSENHOVE, L.N., Analysis of Scheduling Rules for FMS", International Journal of Production Research, 28,785-802, 1990.
- NOF, S. Y., BARASH, M., SOLBERG, J.J., Operational Control of Item Flow in Versatile Manufacturing Systems, International Journal of Production Research, 17, 479-489, 1979.
- NORMAN, V.B., Future Direction in Manufacturing Simulation, Industrial Engineering, pg. 36-37, 1992.
- PANWALKER, S. S., ISKANDER, W., A Survey of Scheduling Rules, Operations Reserach, 25, pg. 45-61, 1977.
- PEGDEN, .C.D. et al., Introduction to Simulation Using SIMAN, McGraw-Hill, New Jersey, 1990

REFERÊNCIAS BIBLIOGRÁFICAS

- PHILIPOOM, P. R., FRY, T. D., The Robustness of Selected Job Shop Dispatching Rules with Respect to Load Balance and Work-Flow Structure, Journal of Operational Research, 41, pg. 897-906, 1990.
- PRISTKER, A.A.B., Simulation: The Premier Technique of Industrial Engineering, Industrial Engineering, pg. 25-26, July 1992.
- RACHAMADUGU, R.V. e STECKE, K.E., Classification and Review of FMS Scheduling Procedures, Working Paper, Graduate School of Business Administration, The University of Michigan, Ann Arbor, 1989.
- RAMAN, N. et al., Real time scheduling of automated manufacturing center, European Journal of Operational Research, 40, 222-224, 1989
- SABUNCUOGLU, I. e HOMMERTZHEIM, D.L., Experimental Investigation of FMS Machine and AGV Scheduling Rules Against the Mean Flow-Time criterion, European Journal of Operational Research, V-30, nº 7, 1617-1635, 1992.
- SCHRIBER, T.J. e STECKE, K.E., Machine Utilizations Achieved Using Balanced FMS Production Ratios in a Simulation Setting, Annals of Productions Research, 32, 579-595, 1988.
- STECKE, K.E., A Hierarchical Approach to Solving Machining Grouping and Loading Problems of FMS, European Journal of Operational Research, 24, 369-378, 1986.
- STECKE, K.E. e SOLBERG, J.J., Loading and Control Policies for FMS, International Journal of Production Research, 19, 481-490, 1981.
- YIM, D. S. e LINN, R. J., Push and Pull Rules for Dispatching AGV's in a FMS, International Journal of Production Research, 31, 43-57, 1993.
- WATANABE, T. et al., Job-Shop Scheduling Using Neural Networks, Control Engineering Practice, V.1, 6, pg. 957-961, 1993.
- WEINBERG, R., Computer Simulation of a Living Cell, Doctoral Dissertation, University of Michigan, USA, 1970.
- WHITNEY, A. e SURI, R., Algorithm for parts and machines selection in FMS, Annals of Operations. Research, 3, 95-117, 1985.

REFERÊNCIAS BIBLIOGRÁFICAS

- WU, S. D., An Expert System Approach for the Control and Scheduling of Flexible Manufacturing Cells, Ph.D Dissertation, Pennsylvania State University, USA, 1987.
- WU, S.D., WYSK, R.A., An Application of Discrete Event Simulation to On-Line Control and Scheduling in Flexible Manufacturing, International Journal of Production Research, 27, 1603-1623, 1989.
- ZEESTRATHEM, M.J., The Look Ahead Dispatching Procedure, International Journal of Production Research, 28, 369-384, 1990.
- ZHUANG, L. e HINDI, K. S., Approximate Decomposition for Closed Queueing Network Models of FMS with Block-and-Wait and State-Dependent Routing Mechanism, European Journal of Operational Research, 67, 3, 373-387, 1993.
- ZIMMERMANN, H. J., Rational Decision Making in Fuzzy Environment and Artificial Intelligence, 1992.